

Università degli Studi di Salerno
Corso di laurea in Informatica

INGEGNERIA DEL SOFTWARE

“StudyMe”

TEST CASE EXECUTION

Docente:

Andrea De Lucia

Studenti:

Nome

Buono Claudia
Esposito Mariarosaria
Tripodi Maria Rachele

Matricola

0512105296
0512105692
0512105356

Anno accademico 2019/20

Sommario

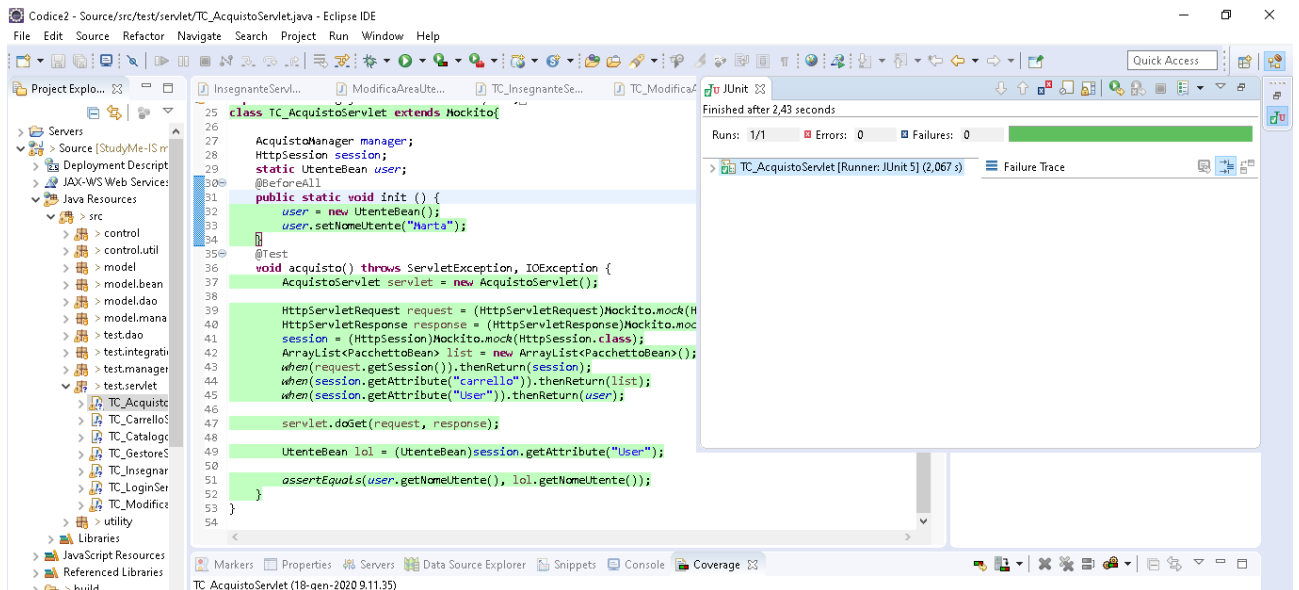
1. Test di unità servlet	4
1.1 AcquistoServlet.....	4
1.2 InsegnanteServlet	4
1.3 GestoreServlet	5
1.4 LoginServlet.....	5
1.5 ModificaAreaUtenteServlet	6
1.6 RecensioneServlet.....	6
2.Test di Unità Manager	7
2.1 AcquistoManager.....	7
2.2 InsegnanteManager	7
2.3 RecensioneManager.....	8
2.4 SottocategoriaManager.....	8
2.5 UtenteManager	8
3.Test di Unità Dao	9
3.1 AcquistoDao.....	9
3.2 GestoreDao	9
3.3 InsegnanteDao	10
3.4 OrdineDao	10
3.5 PacchettoDao	11
3.6 RecensioneDao	11
3.7 SottocategoriaDao	11
3.8UtenteDao	12
4.Test di integrazione Servlet	13
4.1AcquistoServlet.....	13
4.2InsegnanteServlet	13
4.3GestoreServlet	14
4.4LoginServlet.....	14
4.5ModificaAreaUtenteServlet	15
4.6RecensioneServlet	15
5 Test di integrazione Manager	16
5.1 AcquistoManager.....	16
5.2 InsegnanteManager	16
5.3 RecensioneManager.....	17
5.4 SottocategoriaManager.....	17
5.5 UtenteManager	17
6.Introduzione.....	18
7.Gestione Utente.....	19

7.1 Login	19
7.2 Registrazione.....	22
7.3 Cambio email.....	28
7.4 Cambio password	31
8. Gestione pacchetto	34
8.1 Crea Pacchetto	34
8.2 Modifica pacchetto	47
9. Gestione Lezioni.....	53
9.1 Inserisci lezione	53
9.2 Modifica Lezione	59
9.3 Inserimento recensione	63

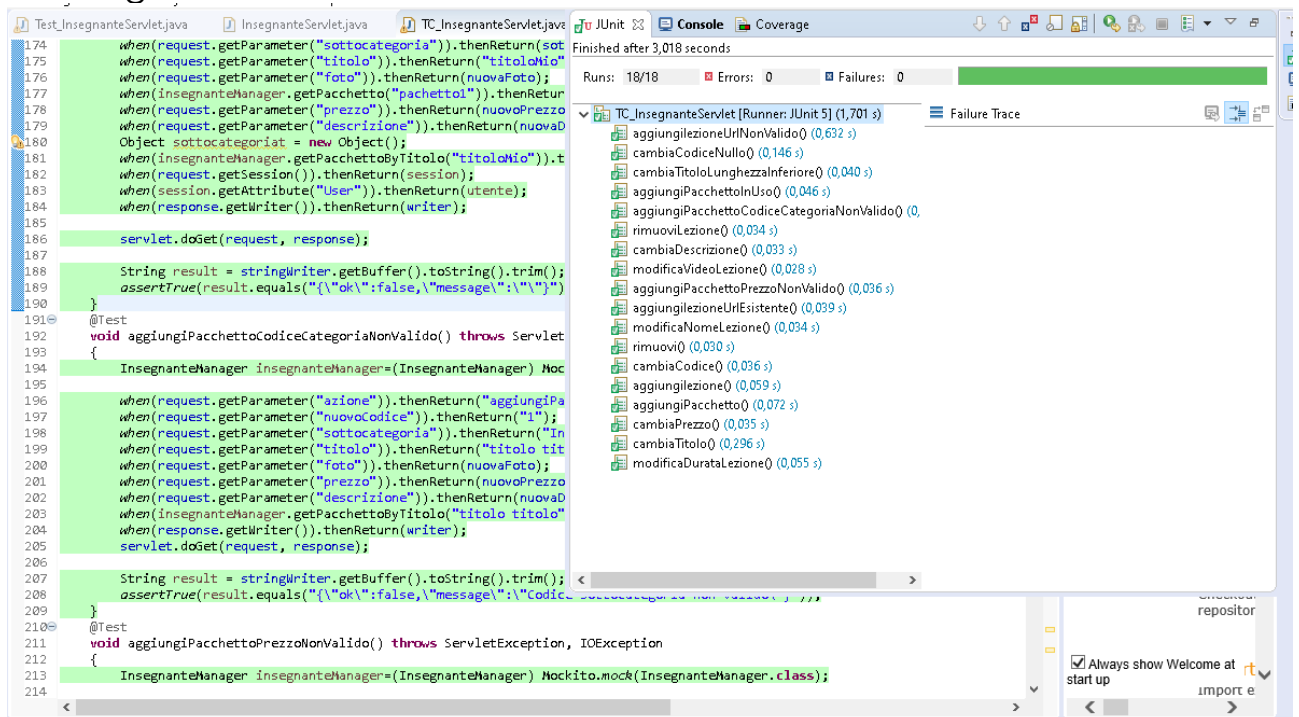
TEST JUNIT

1. Test di unità servlet

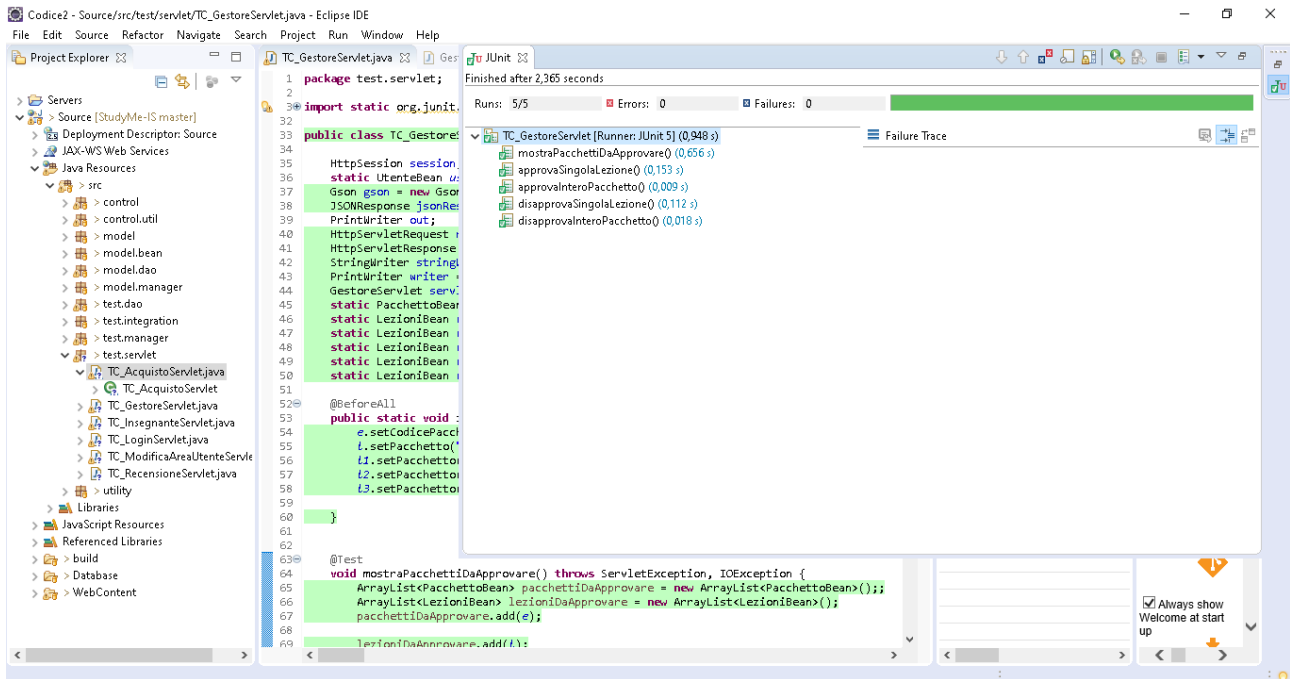
1.1 AcquistoServlet



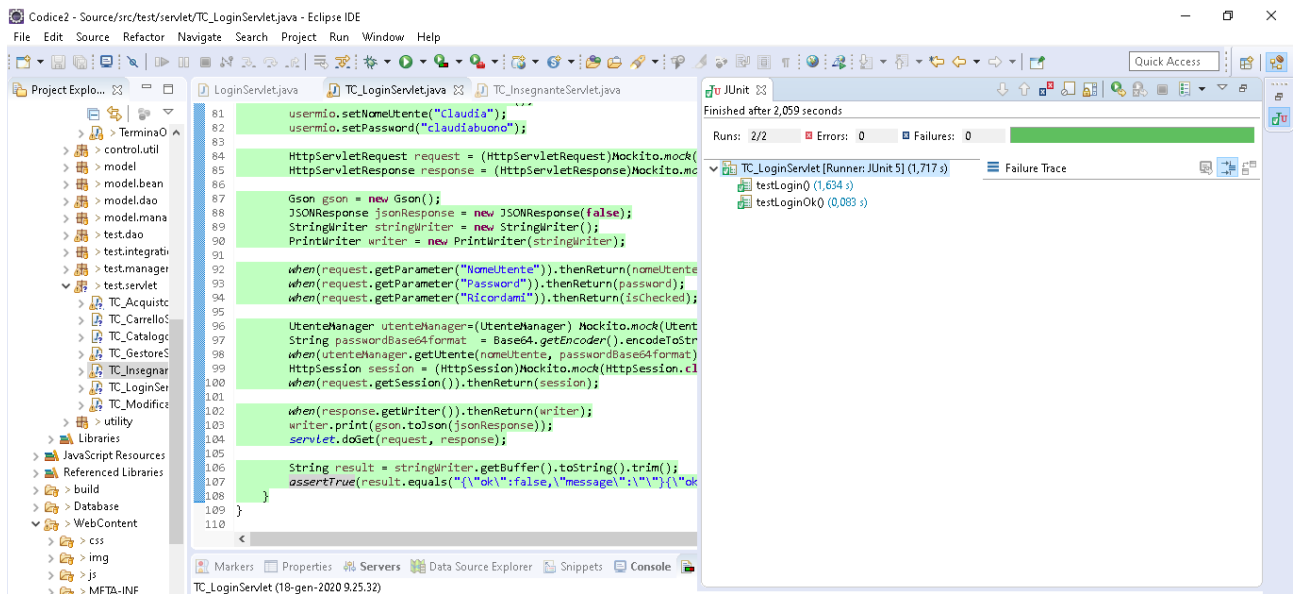
1.2 InsegnanteServlet



1.3 GestoreServlet



1.4 LoginServlet



1.5 ModificaAreaUtenteServlet

The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Displays the project structure, including packages like `control`, `model`, `model.bean`, `model.dao`, `model.manager`, `test.dao`, `test.integration`, `test.manager`, `test.servlet`, and `utility`.
- Editor:** Shows the implementation of `ModificaAreaUtenteServlet.java`. The code includes a `doGet` method that handles GET requests and a `doPost` method that handles POST requests. It uses `HttpSession` to manage user sessions and `StringWriter` to write responses.
- JUnit:** Shows the JUnit test results for `TC_ModificaAreaUtenteServlet`. The tests passed, with a total of 2,351 instructions executed.

1.6 RecensioneServlet

The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Displays the project structure, including packages like `control`, `model`, `model.bean`, `model.dao`, `model.manager`, `test.dao`, `test.integration`, `test.manager`, `test.servlet`, and `utility`.
- Editor:** Shows the implementation of `RecensioneServlet.java`. The code includes a `doGet` method that handles GET requests and a `doPost` method that handles POST requests. It uses `HttpSession` to manage user sessions and `StringWriter` to write responses.
- JUnit:** Shows the JUnit test results for `TC_RecensioneServlet`. The tests passed, with a total of 2,095 instructions executed.

2. Test di Unità Manager

2.1 AcquistoManager

```
package test.manager;

import static org.junit.Assert.assertNotNull;

class TC_AcquistoManager extends Mockito{

    AcquistoManager manager;
    OrdineDao ordineMock;
    AcquistoDao acquistoMock;

    @Test
    void testGetDataOdierna() {
        manager = new AcquistoManager();

        String data;
        data = manager.getDataOdierna();
        assertNotNull(data);
    }

    @Test
    void testGetOrdine() {
        ordineMock = (OrdineDao)Mockito.mock(OrdineDao.class);
        acquistoMock = (AcquistoDao)Mockito.mock(AcquistoDao.class);
        ArrayList<PacchettoBean> carrello = new ArrayList<PacchettoBean>();
        PacchettoBean pacchetto = new PacchettoBean();
        carrello.add(pacchetto);

        OrdineBean ordine = new OrdineBean();
        AcquistoBean acquisto = new AcquistoBean();

        when(ordineMock.insert(ordine)).thenReturn(1);
        manager = new AcquistoManager();
        manager.setDao(acquistoMock);
        manager.setDaoOrdine(ordineMock);
        manager.getOrdine("Damiana", carrello);
        verify(acquistoMock).insertAcquisto(acquisto);
    }
}
```

Finished after 1,281 seconds

Runs: 2/2 Errors: 0 Failures: 0

> TC_AcquistoManager [Runner: JUnit 5] (1,094 s)

Failure Trace

2.2 InsegnanteManager

```
package test.manager;

import static org.junit.Assert.assertEquals;

class InsegnanteManagerTest extends Mockito {

    InsegnanteDao insegnanteMock;
    PacchettoDao pacchettoMock;
    InsegnanteManager insegnanteManager;

    @Test
    void testInserimentoPacchetto() {
        insegnanteMock = (InsegnanteDao)Mockito.mock(InsegnanteDao.class);
        insegnanteManager = new InsegnanteManager();

        PacchettoBean pacchettoMock = new PacchettoBean();
        pacchettoMock.setCodicePacchetto("pac333");
        pacchettoMock.setDescrizione("Corso sulla gestione dei processi");
        pacchettoMock.setSottocategoria("Sistema operativo");
        pacchettoMock.setPrezzo(33);
        pacchettoMock.setTitolo("Processi");
        pacchettoMock.setFoto("https://cdn01.alison-static.net/courses/1052/alison_courseware_intro_1052.jpg");

        PacchettoBean pacchetto = new PacchettoBean();

        when(insegnanteMock.inserPacchetto("pac333", "Marianosaria", "inf001", 33, "Corso sulla gestione process
        pacchetto = insegnanteManager.inserPacchetto("pac333", "Marianosaria", "inf001", 33, "Corso sulla gestio
        assertEquals("pac333", pacchetto.getCodicePacchetto());
    }

    @Test
    void testGetPacchetto() {
        pacchettoMock = (PacchettoDao)Mockito.mock(PacchettoDao.class);
        insegnanteManager = new InsegnanteManager();

        PacchettoBean pacchettoFake = new PacchettoBean();
        pacchettoFake.setCodicePacchetto("pac001");

        when(pacchettoMock.getPacchetto("pac001")).thenReturn(pacchettoFake);
    }
}
```

Finished after 2,43 seconds

Runs: 16/16 Errors: 0 Failures: 0

InsegnanteManagerTest [Runner: JUnit 5] (2,220 s)

- testDeletePacchetto() (1,386 s)
- testGetPacchettoByTitolo() (0,728 s)
- testGetPacchetto() (0,015 s)
- testUpdatePrice() (0,003 s)
- testUpdateTitle() (0,002 s)
- testUpdateCode() (0,004 s)
- testUpdateDesc() (0,002 s)
- testDeleteLesson() (0,002 s)
- testGetAcquirenti() (0,013 s)
- testGetLezioniByUrl() (0,010 s)
- testUpdateUriLesson() (0,002 s)
- testUpdateDurationLesson() (0,002 s)
- testGetLezioniByTitle() (0,009 s)

Failure Trace

2.3 RecensioneManager

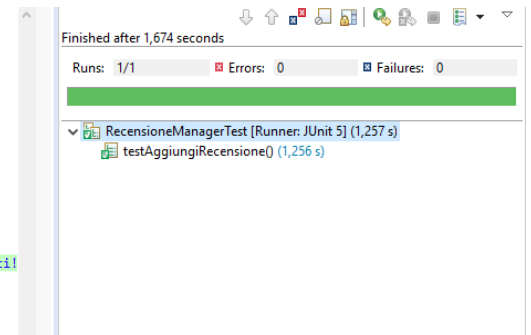
```
package test.manager;

import org.junit.jupiter.api.Test;

class RecensioneManagerTest extends Mockito {

    RecensioneDao recensioneMock;
    RecensioneManager recensioneManager;

    @Test
    void testAggiungiRecensione() {
        recensioneMock = (RecensioneDao)Mockito.mock(RecensioneDao.class);
        RecensioneManager manager = new RecensioneManager();
        manager.setDao(recensioneMock);
        manager.aggiungiRecensione("Damiana", "pac001", "Ottimo corso", "Corso consigliato a tutti!");
        verify(recensioneMock).aggiungiRecensione("Damiana", "pac001", "Ottimo corso", "Corso consigliato a tutti!");
    }
}
```



2.4 SottocategoriaManager

```
package test.manager;

import static org.junit.jupiter.api.Assertions.*;

class TC_SottocategoriaManager extends Mockito {

    SottocategoriaDao sottocategoriaMock;
    SottocategoriaManager sottocategoria;

    @Test
    void selezionaSottocategorieInsegnanteTest() {
        sottocategoriaMock = (SottocategoriaDao)Mockito.mock(SottocategoriaDao.class);
        sottocategoria = new SottocategoriaManager();

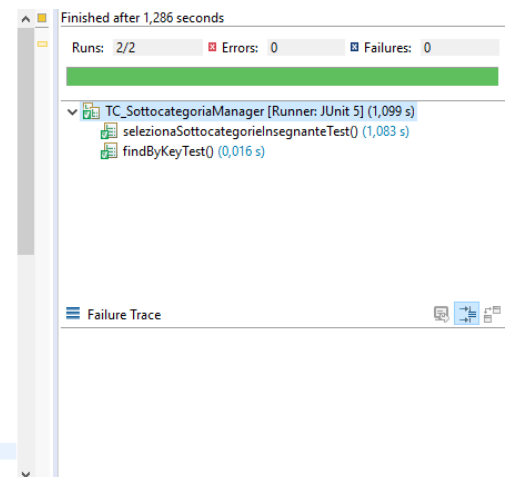
        ArrayList<SottocategoriaBean> sottocategorieMock = new ArrayList<SottocategoriaBean>();

        SottocategoriaBean s = new SottocategoriaBean();
        s.setIdSottoCat("inf001");
        s.setNomeSott("Sistema operativo");

        SottocategoriaBean s2 = new SottocategoriaBean();
        s.setIdSottoCat("inf002");
        s.setNomeSott("Reti");

        sottocategorieMock.add(s);
        sottocategorieMock.add(s2);

        ArrayList<SottocategoriaBean> sottocategorie = new ArrayList<SottocategoriaBean>();
    }
}
```



2.5 UtenteManager

```
UtenteManagerTest.java CatalogoServlet.java

package test.manager;

import static org.junit.jupiter.api.Assertions.*;

class UtenteManagerTest extends Mockito {

    GestoreDao gestoreMock;
    UtenteDao utenteMock;
    UtenteManager utenteManager;

    @Test
    void testLogin() throws IOException {
        utenteMock = (UtenteDao)Mockito.mock(UtenteDao.class);
        utenteManager = new UtenteManager();
        String password = "cuoca";
        String passwordBase64format = Base64.getEncoder().encodeToString(password.getBytes());
        UtenteBean utenteLoggato = new UtenteBean();
        utenteLoggato.setNomeUtente("Martina");
        utenteLoggato.setPassword(passwordBase64format);

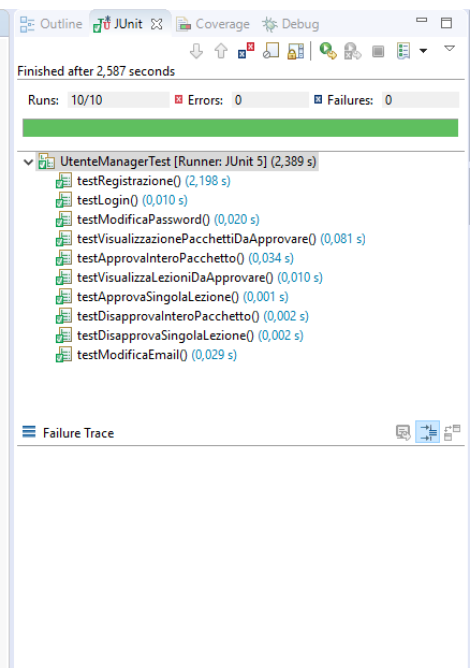
        UtenteBean utenteM = new UtenteBean();

        when(utenteMock.login("Martina", passwordBase64format)).thenReturn(utenteLoggato);
        utenteM = utenteManager.login("Martina", passwordBase64format);
        assertEquals(utenteM.getNomeUtente(), utenteLoggato.getNomeUtente());
    }

    @Test
    void testModificaEmail() throws IOException {
        utenteMock = (UtenteDao)Mockito.mock(UtenteDao.class);
        utenteManager = new UtenteManager();

        boolean res;
        when(utenteMock.updateEmail("mesposito@gmail.com", "chiccaesp98@gmail.com")).thenReturn(true);
        res = utenteManager.setEmail("mesposito@gmail.com", "chiccaesp98@gmail.com");
        assertEquals(res, true);
    }

    @Test
    // ...
}
```



3. Test di Unità Dao

3.1 AcquistoDao

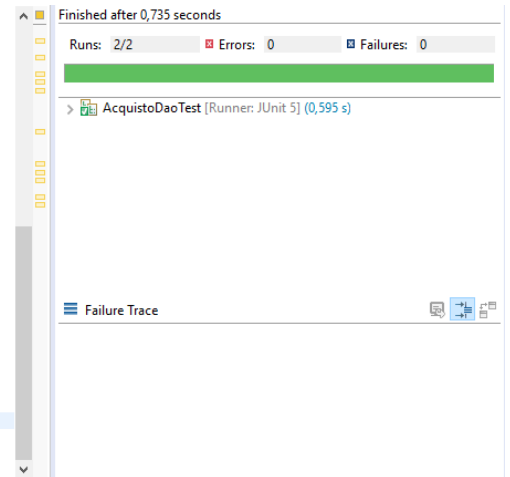
```
class AcquistoDaoTest {
    AcquistoDao acquisto;

    @BeforeEach
    void setUp() throws Exception {
        acquisto = new AcquistoDao();
    }

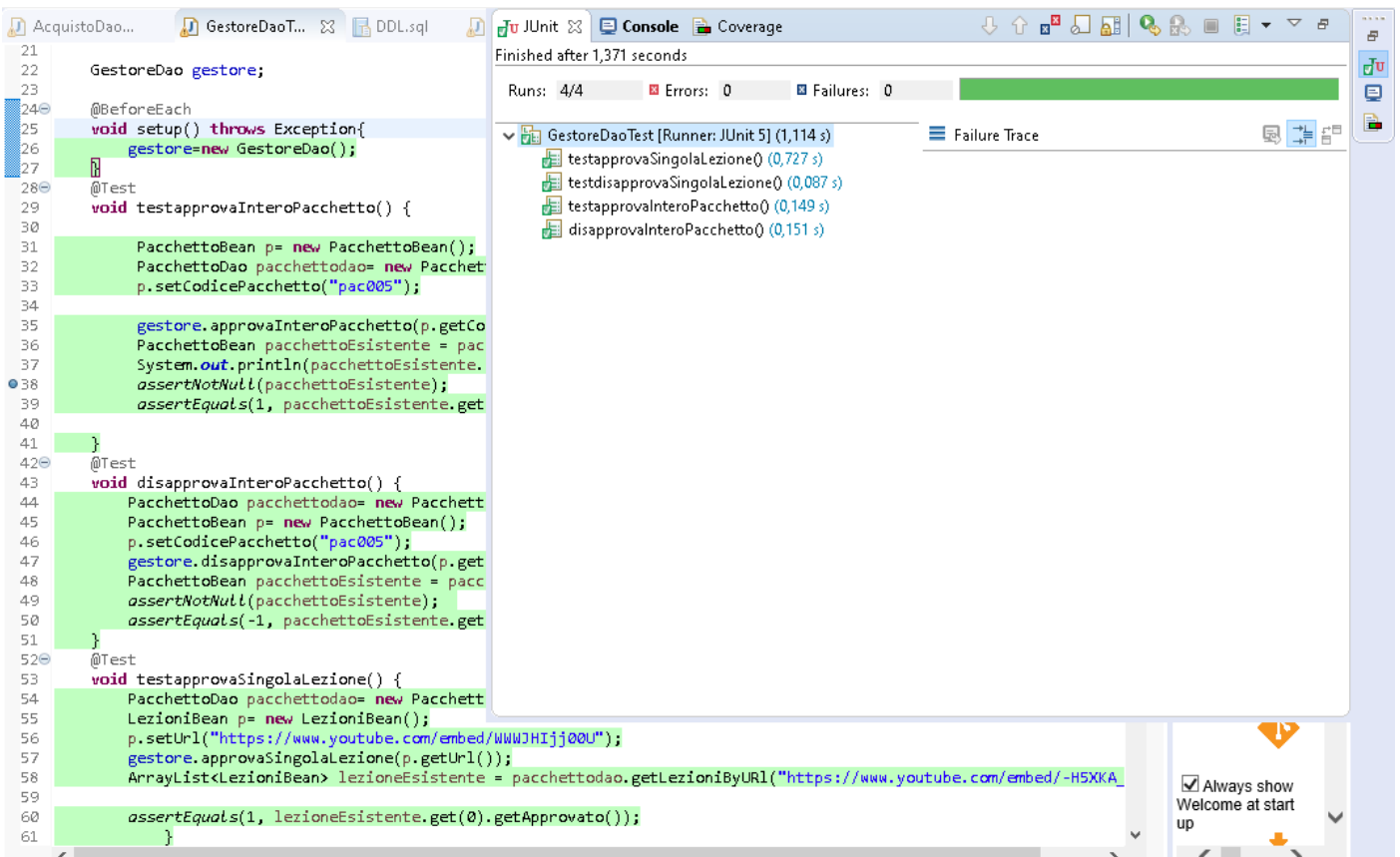
    @Test
    void testInsertAcquisto() {
        AcquistoBean bean = new AcquistoBean();
        bean.setCodiceP("pac004");
        bean.setImporto(12.8);
        bean.setNumOrdine(1);
        bean.setTitoloPacchetto("Impara Python programming");
        acquisto.insertAcquisto(bean);

        Boolean res = acquisto.existingPurchase(bean.getNumOrdine());
        assertTrue(res);
    }

    @Test
    void testEsistenzaAcquisto() {
        Boolean res = acquisto.existingPurchase(1);
        assertTrue(res);
    }
}
```



3.2 GestoreDao



3.3 InsegnanteDao

The screenshot shows the IDE with the `InsegnanteDaoTest.java` file open. The code contains several JUnit tests for the `InsegnanteDao` interface. The right-hand pane displays the JUnit test results, showing that all tests passed successfully.

```
119 @Test
120 void testUpdateTitleLesson() {
121     LezioniBean p = new LezioniBean();
122     p.setTitolo("Facebook per il business : Differenziale");
123     insegnante.updateTitleLesson(p.getTitolo(), "nuovo");
124     ArrayList<LezioniBean> lezioneEsistente = insegnante.getLezioni();
125     assertNotNull(lezioneEsistente);
126 }
127
128 @Test
129 void testUpdateUrlLesson() {
130     LezioniBean p = new LezioniBean();
131     p.setUrl("https://www.youtube.com/embed/05C9JNZ");
132     insegnante.updateUrlLesson("https://www.youtube.com/watch?v=05C9JNZ", "nuovo");
133     ArrayList<LezioniBean> lezioneEsistente = insegnante.getLezioni();
134     assertNotNull(lezioneEsistente);
135 }
136
137 @Test
138 void testUpdateDurationLesson() {
139     LezioniBean p = new LezioniBean();
140     p.setTitolo("Facebook per il business : Differenziale");
141     insegnante.updateDurationLesson(p.getTitolo(), "11:11");
142     ArrayList<LezioniBean> lezione = insegnante.getLezioni();
143     assertEquals("11:11", lezione.get(0).getDurata());
144 }
145
146 @Test
147 void testGetOrdine() {
148     String nomeCliente = "Pasquale";
149     List<OrdineBean> ordini = insegnante.getOrdine(nomeCliente);
150     assertNotNull(ordini);
151 }
152
153 }
```

JUnit Results: Finished after 18,493 seconds. Runs: 11/11, Errors: 0, Failures: 0.

- testdeletePacchetto() (17,306 s)
- testUpdateDesc() (0,328 s)
- testInsertPacchetto() (0,062 s)
- testupdateTitolo() (0,063 s)
- testUpdateUrlLesson() (0,046 s)
- testUpdateDurationLesson() (0,032 s)
- testUpdateTitleLesson() (0,047 s)
- testupdateCode() (0,234 s)
- testGetOrdine() (0,031 s)
- testupdatePrice() (0,031 s)
- testInsertLesson() (0,047 s)

3.4 OrdineDao

The screenshot shows the IDE with the `OrdineDaoTest.java` file open. The code contains a JUnit test for the `OrdineDao` interface. The right-hand pane displays the JUnit test results, showing that the test passed successfully.

```
1 package test.dao;
2
3 import static org.junit.Assert.assertEquals;
4 import static org.junit.Assert.assertFalse;
5 import static org.junit.Assert.assertTrue;
6 import static org.junit.Assert.fail;
7
8 import java.sql.Date;
9
10 import org.junit.jupiter.api.BeforeEach;
11 import org.junit.jupiter.api.Test;
12
13 import model.bean.OrdineBean;
14 import model.dao.OrdineDao;
15
16 class OrdineDaoTest {
17     OrdineDao ordine;
18
19     @BeforeEach
20     void setup() throws Exception {
21         ordine = new OrdineDao();
22     }
23
24     @Test
25     void testinsert() {
26         Date d = new Date(12, 12, 12);
27
28         OrdineBean ordine = new OrdineBean();
29         ordine.setCliente("Damiana");
30         ordine.setNumOrdine(1);
31         ordine.setData(d);
32
33         OrdineDao dao = new OrdineDao();
34         int n = dao.insert(ordine);
35
36         assertEquals(-1, n);
37     }
38 }
39
40 }
```

JUnit Results: Finished after 0,93 seconds. Runs: 1/1, Errors: 0, Failures: 0.

- testinsert() (0,661 s)

3.5 PacchettoDao

```
package test.dao;

import static org.junit.Assert.assertEquals;

class PacchettoDaoTest {

    PacchettoDao pacchetto;

    @BeforeEach
    void setUp() throws Exception {
        pacchetto = new PacchettoDao();
    }

    @Test
    void testGetAllPacchetti() {
        List<PacchettoBean> pacchetti = new ArrayList<PacchettoBean>();
        pacchetti = pacchetto.getAllPacchetti();
        assertNotNull(pacchetti);
    }

    @Test
    void testGetPacchetto() {
        String codiceP = "pac001";

        PacchettoBean pac = new PacchettoBean();
        pac = pacchetto.getPacchetto(codiceP);

        assertNotNull(pac);
    }
}
```

Finished after 1,116 seconds

Runs: 14/14 Errors: 0 Failures: 0

PacchettoDaoTest [Runner: JUnit 5] (0,949 s)

- testDeletePacchetto() (0,664 s)
- testSearchPackage() (0,010 s)
- testGetPacchettoByTitolo() (0,009 s)
- testGetPacchetto() (0,009 s)
- testGetLezioniApprovate() (0,016 s)
- testGetCategoriaRaggruppatoApprovato() (0,112 s)
- testGetAllPacchetti() (0,010 s)
- testGetAcquirenti() (0,007 s)
- testGetLezioniByTitolo() (0,007 s)

Failure Trace

3.6 RecensioneDao

```
package test.dao;

import static org.junit.Assert.assertNotNull;

class RecensioneDaoTest {

    RecensioneDao recensione;

    @BeforeEach
    void setUp() throws Exception {
        recensione = new RecensioneDao();
    }

    @Test
    void testAggiungiRecensione() {
        String nomeUtente = "Damiana";
        String codicePacchetto = "pac002";
        String titoloRecensione = "Programmazione";
        String testoRecensione = "Questo corso è stato interessante";

        recensione.aggiungiRecensione(nomeUtente, codicePacchetto, titoloRecensione, testoRecensione);
        ArrayList<RecensioneBean> recensioni = new ArrayList<RecensioneBean>();

        PacchettoDao p = new PacchettoDao();
        recensioni = p.getRecensioni("pac002");

        for (RecensioneBean r : recensioni) {
            if (r.getCliente().equals("Damiana")) {
                // ...
            }
        }
    }
}
```

Finished after 0,988 seconds

Runs: 2/2 Errors: 0 Failures: 0

RecensioneDaoTest [Runner: JUnit 5] Failure Trace

- testAggiungiRecensione() (0,688 s)
- isAlreadyReviewed() (0,027 s)

Failure Trace

3.7 SottocategoriaDao

```
package test.dao;

import static org.junit.jupiter.api.Assertions.assertEquals;

class SottocategoriaDaoTest {

    SottocategoriaDao sottocategoria;

    @BeforeEach
    void setUp() throws Exception {
        sottocategoria = new SottocategoriaDao();
    }

    @Test
    void testFindByKey() throws SQLException {
        Object codiceS = "fot003";

        SottocategoriaBean s = new SottocategoriaBean();
        s = sottocategoria.findByKey(codiceS);
        assertEquals(codiceS, s.getIdSottoCat());
    }

    @Test
    void testSelezionaSottocategorieInsegnante() {
        String utente = "Rachele";

        ArrayList<SottocategoriaBean> sottocategorie = new ArrayList<SottocategoriaBean>();
        sottocategorie = sottocategoria.selezionaSottocategorieInsegnante(utente);

        assertNotNull(sottocategorie);
    }
}
```

Finished after 0,83 seconds

Runs: 2/2 Errors: 0 Failures: 0

SottocategoriaDaoTest [Runner: JUnit 5] (0,640 s)

- testSelezionaSottocategorieInsegnante() (0,616 s)
- testFindByKey() (0,024 s)

Failure Trace

3.8 UtenteDao

```
package test.dao;

import static org.junit.Assert.assertEquals;

class UtenteDaoTest {
    UtenteDao utente;
    String email;

    @BeforeEach
    void setUp() throws Exception{
        utente = new UtenteDao();
    }

    @Test
    void testLogin() {
        String nomeUtente = "Claudia";
        String password = "claudiabuono";
        String passwordBase64format = Base64.getEncoder().encodeToString(password.getBytes());

        UtenteBean u = utente.login(nomeUtente, passwordBase64format);
        assertNotNull(u);
        assertEquals(nomeUtente, u.getNomeUtente());
    }

    @Test
    void testRegistrazione() {
        boolean resp = utente.registration("eugenio@live.it", "Eugenio", "Eugenio98");
        assertEquals(true, resp);
    }

    @Test
    void testModificaPassword() {
        String passwordNuova = "Mariarosaria";
        String passwordBase64format = Base64.getEncoder().encodeToString(passwordNuova.getBytes());
        boolean resp = utente.updatePassword("mesposito@gmail.com", passwordBase64format);
        assertEquals(true, resp);
    }
}
```

Finished after 1,727 seconds

Runs: 5/5 Errors: 0 Failures: 0

▼ UtenteDaoTest [Runner: JUnit 5] (1,349 s)

- testRegistrazione() (1,248 s)
- testLogin() (0,038 s)
- testModificaPassword() (0,032 s)
- testGetAllAcquirenti() (0,010 s)
- testModificaEmail() (0,020 s)

Failure Trace

4. Test di integrazione Servlet

4.1 AcquistoServlet

The screenshot shows an IDE with a Java test class `TestAcquistoServlet.java` and its execution results in the JUnit console.

Test Class Code:

```
14 import org.junit.jupiter.api.BeforeAll;
15 import org.junit.jupiter.api.Test;
16 import org.mockito.Mockito;
17
18 import control.AcquistoServlet;
19 import model.bean.PacchettoBean;
20 import model.bean.UtenteBean;
21 import model.manager.AcquistoManager;
22
23 class Test_AcquistoServlet extends Mockito{
24
25     AcquistoManager manager;
26     HttpSession session;
27     static UtenteBean user;
28     @BeforeAll
29     public static void init () {
30         user = new UtenteBean();
31         user.setNomeUtente("Marta");
32     }
33     @Test
34     void acquisto() throws ServletException, IOException {
35         AcquistoServlet servlet = new AcquistoServlet();
36
37         HttpServletRequest request = (HttpServletRequest)Mockito.mock(HttpServletReq
38         HttpServletResponse response = (HttpServletResponse)Mockito.mock(HttpServlet
39         session = (HttpSession)Mockito.mock(HttpSession.class);
40         ArrayList<PacchettoBean> list = new ArrayList<PacchettoBean>();
41         when(request.getSession()).thenReturn(session);
42         when(session.getAttribute("carrello")).thenReturn(list);
43         when(session.getAttribute("User")).thenReturn(user);
44
45         servlet.doGet(request, response);
46
47         UtenteBean lol = (UtenteBean)session.getAttribute("User");
48         assertEquals(user.getNomeUtente(), lol.getNomeUtente());
49     }
50 }
51 }
```

JUnit Console:

Finished after 2,115 seconds

Runs: 1/1 Errors: 0 Failures: 0

Test_AcquistoServlet [Runner: JUnit 5] (1,794 s)

acquisto() (1,794 s)

4.2 InsegnanteServlet

The screenshot shows an IDE with a Java test class `TestInsegnanteServlet.java` and its execution results in the JUnit console.

Test Class Code:

```
255 void modificaNomeLezione() throws ServletException, IOException {
256     String nuovoNomeLezione="corso foto";
257     when(request.getParameter("nuovoNomeLezione")).thenReturn(n
258     when(response.getWriter()).thenReturn(writer);
259     servlet.doGet(request, response);
260
261     String result = stringWriter.getBuffer().toString().trim();
262     assertNotNull(result);
263 }
264 @Test
265 void modificaVideoLezione() throws ServletException, IOException {
266     String nuovoUrlLezione="https://www.youtube.com/embed/2s0d0
267     when(request.getParameter("nuovoUrlLezione")).thenReturn(nu
268     when(response.getWriter()).thenReturn(writer);
269     servlet.doGet(request, response);
270
271     String result = stringWriter.getBuffer().toString().trim();
272     assertNotNull(result);
273 }
274 @Test
275 void modificaDurataLezione() throws ServletException, IOException {
276     String nuovaDurata= "11:45";
277     when(request.getParameter("nuovaDurataLezione")).thenReturn
278     when(response.getWriter()).thenReturn(writer);
279     servlet.doGet(request, response);
280
281     String result = stringWriter.getBuffer().toString().trim();
282     assertNotNull(result);
283 }
284 @Test
285 void rimuoviLezione() throws ServletException, IOException {
286     when(response.getWriter()).thenReturn(writer);
287     servlet.doGet(request, response);
288
289     String result = stringWriter.getBuffer().toString().trim();
290     assertNotNull(result);
291 }
292 }
293 }
294 }
295 }
```

JUnit Console:

Finished after 3,221 seconds

Runs: 18/18 Errors: 0 Failures: 0

Test_InsegnanteServlet [Runner: JUnit 5] (1,839 s)

- aggiungiLezioneUrlNonValido() (0,721 s)
- cambiaCodiceNull() (0,072 s)
- cambiaTitoloLunghezzaInferiore() (0,056 s)
- aggiungiPacchettoInUso() (0,043 s)
- aggiungiPacchettoCodiceCategoriaNonValido() (0,036 s)
- rimuoviLezione() (0,036 s)
- cambiaDescrizione() (0,038 s)
- modificaVideoLezione() (0,030 s)
- aggiungiPacchettoPrezzoNonValido() (0,035 s)
- aggiungiLezioneUrlEsistente() (0,036 s)
- modificaNomeLezione() (0,028 s)
- rimuovi() (0,028 s)
- cambiaCodice() (0,035 s)
- aggiungiLezione() (0,009 s)
- aggiungiPacchetto() (0,066 s)
- cambiaPrezzo() (0,033 s)
- cambiaTitolo() (0,386 s)
- modificaDurataLezione() (0,057 s)

4.3 GestoreServlet

The screenshot shows an IDE with two tabs: `Test_GestoreServlet.java` and `TC_GestoreServlet.java`. The `Test_GestoreServlet.java` file contains the following code:

```
55 }
56
57
58 @Test
59 void mostraPacchettiDaApprovare() throws ServletException,
60     ArrayList<PacchettoBean> pacchettiDaApprovare = new Arr
61     ArrayList<LezioniBean> lezioniDaApprovare = new ArrayLi
62     pacchettiDaApprovare.add(e);
63
64     lezioniDaApprovare.add(l);
65     lezioniDaApprovare.add(l1);
66     lezioniDaApprovare.add(l2);
67     lezioniDaApprovare.add(l3);
68     lezioniDaApprovare.add(l4);
69     when(request.getParameter("action")).thenReturn("mostra
70     when(response.getWriter()).thenReturn(writer);
71
72     servlet.doGet(request, response);
73     String result = stringWriter.getBuffer().toString().tri
74     System.out.println(result);
75     assertTrue(result.equals("{\"ok\":true,\"message\":\"ok
76 }
77
78 @Test
79 void approvaInteroPacchetto() throws ServletException, IOEx
80     when(request.getParameter("action")).thenReturn("approv
81     when(request.getParameter("codicePacchetto")).thenRetur
82     when(response.getWriter()).thenReturn(out);
83     when(response.getWriter()).thenReturn(writer);
84     servlet.doGet(request, response);
85
86     String result = stringWriter.getBuffer().toString().tri
87     assertTrue(result.equals("{\"ok\":true,\"message\":\"
88 }
89
90 @Test
91 void disapprovaInteroPacchetto() throws ServletException, IOEx
92     when(request.getParameter("action")).thenReturn("disapprovaInteroPacchetto");
93     when(request.getParameter("codicePacchetto")).thenReturn("pac121");
94     when(response.getWriter()).thenReturn(out);
```

The right-hand side of the IDE shows the JUnit console window. It indicates that the tests finished after 2,263 seconds. The summary shows: Runs: 5/5, Errors: 0, Failures: 0. A green progress bar is visible. Below the summary, the test `Test_GestoreServlet [Runner: JUnit 5] (0,939 s)` is listed with a 'Failure Trace' link.

4.4 LoginServlet

The screenshot shows an IDE with three tabs: `TC_LoginServlet.java`, `Test_LoginServlet.java`, and `TestUtenteManager.java`. The `Test_LoginServlet.java` file contains the following code:

```
65     when(response.getWriter()).thenReturn(writer);
66     writer.print(gson.toJson(jsonResponse));
67     servlet.doGet(request, response);
68
69     UtenteBean lol = (UtenteBean)session.getAttribute("User
70     assertEquals(user1.getNomeUtente(), null);
71 }
72
73 @Test
74 public void testLoginOk() throws Exception {
75     servlet = new LoginServlet();
76     String isChecked= "true";
77     UtenteBean usermio= new UtenteBean();
78     usermio.setNomeUtente("Rachele");
79     usermio.setPassword("mangiona");
80
81     HttpServletRequest request = (HttpServletRequest)Mockit
82     HttpServletResponse response = (HttpServletResponse)Moc
83
84     Gson gson = new Gson();
85     JsonResponse jsonResponse = new JsonResponse(false);
86     StringWriter stringWriter = new StringWriter();
87     PrintWriter writer = new PrintWriter(stringWriter);
88
89     when(request.getParameter("NomeUtente")).thenReturn(nom
90     when(request.getParameter("Password")).thenReturn(passw
91     when(request.getParameter("Ricordami")).thenReturn(isCh
92
93     HttpSession session = (HttpSession)Mockito.mock(HttpSes
94     when(request.getSession()).thenReturn(session);
95
96     when(response.getWriter()).thenReturn(writer);
97     servlet.doGet(request, response);
98
99     String result = stringWriter.getBuffer().toString().tri
100     System.out.println(result);
101     assertTrue(result.equals("{\"ok\":true,\"message\":\"OK\", \"content\": \"Rachele\"}"));
102 }
103 }
```

The right-hand side of the IDE shows the JUnit console window. It indicates that the tests finished after 2,052 seconds. The summary shows: Runs: 2/2, Errors: 0, Failures: 0. A green progress bar is visible. Below the summary, the test `Test_LoginServlet [Runner: JUnit 5] (1,721 s)` is listed with a 'Failure Trace' link. Underneath, two sub-tests are shown: `testLoginOk (1,645 s)` and `testLoginOk (0,076 s)`.

4.5 ModificaAreaUtenteServlet

The screenshot displays the Eclipse IDE with the file `Test_ModificaAreaUtenteServlet.java` open. The code is a JUnit test class that tests the `ModificaAreaUtenteServlet`. It includes two test methods: `modificaEmail()` and `modificaPassword()`. Both methods use Mockito to mock the `HttpServletRequest`, `HttpServletResponse`, `HttpSession`, and `ServletContext` objects. The `modificaEmail()` test sets up a request with parameters for email, password, and confirmation, and expects a JSON response with status "ok" and message "OK". The `modificaPassword()` test sets up a request with parameters for email, password, and confirmation, and expects a JSON response with status "ok" and message "OK".

```
10 }
11
12
13 @Test
14 void modificaEmail() throws ServletException, IOException {
15     String nuovaEmailUtente = "claudia@alice.it";
16     String emailUtente="claudiabuono99@gmail.com";
17     String nuovaPasswordUtente = null;
18     String confermaNuovaPasswordUtente=null;
19     when(request.getParameter("NuovaEmailUtente")).thenReturn
20     when(request.getParameter("NuovaPasswordUtente")).thenRe
21     when(request.getParameter("ConfermaNuovaPasswordUtente")
22     HttpSession session = (HttpSession)Mockito.mock(HttpSess
23     when(request.getSession()).thenReturn(session);
24     when(session.getAttribute("User")).thenReturn(loggedUser
25     when(response.getWriter()).thenReturn(writer);
26
27     servlet.doGet(request, response);
28     String result = stringWriter.getBuffer().toString().trim
29     System.out.println(result);
30     assertTrue(result.equals("{\"ok\":true,\"message\":\"OK\"
31 }
32
33 @Test
34 void modificaPassword() throws ServletException, IOException
35 String nuovaEmailUtente = null;
36 String emailUtente=null;
37 String nuovaPasswordUtente = "claudia99";
38 String confermaNuovaPasswordUtente="claudia99";
39 when(request.getParameter("NuovaEmailUtente")).thenReturn
40 when(request.getParameter("NuovaPasswordUtente")).thenRe
41 when(request.getParameter("ConfermaNuovaPasswordUtente")
42 HttpSession session = (HttpSession)Mockito.mock(HttpSess
43 when(request.getSession()).thenReturn(session);
44 when(session.getAttribute("User")).thenReturn(loggedUser
45 when(response.getWriter()).thenReturn(writer);
46 writer.print(gson.toJson(jsonResponse));
47
48 servlet.doGet(request, response);
49 String result = stringWriter.getBuffer().toString().trim();
```

The JUnit test results are shown in the right-hand pane. The test suite `Test_ModificaAreaUtenteServlet` (Runner: JUnit) finished after 2,34 seconds. It contains two tests: `modificaEmail()` (0,903 s) and `modificaPassword()` (0,035 s). Both tests passed successfully.

4.6 RecensioneServlet

The screenshot displays the Eclipse IDE with the file `Test_RecensioneServlet.java` open. The code is a JUnit test class that tests the `RecensioneServlet`. It includes a single test method: `recensione()`. This method uses Mockito to mock the `HttpServletRequest`, `HttpServletResponse`, and `HttpSession` objects. It also creates a `Gson` object and a `JSONResponse` object. The `recensione()` test sets up a request with parameters for user, package code, review title, and review text, and expects a JSON response with status "ok" and message "OK".

```
1 import javax.servlet.ServletException;
2 import javax.servlet.http.HttpServletRequest;
3 import javax.servlet.http.HttpServletResponse;
4
5 import org.junit.jupiter.api.Test;
6 import org.mockito.Mockito;
7
8 import com.google.gson.Gson;
9 import control.RecensioneServlet;
10 import control.util.JSONResponse;
11
12 class Test_RecensioneServlet extends Mockito{
13
14     @Test
15     void recensione() throws ServletException, IOException {
16         HttpServletRequest request = (HttpServletRequest)Mockit
17         HttpServletResponse response = (HttpServletResponse)Mock
18         Gson gson = new Gson();
19         JSONResponse jsonResponse = new JSONResponse(true);
20         StringWriter stringWriter = new StringWriter();
21         PrintWriter writer = new PrintWriter(stringWriter);
22         RecensioneServlet servlet= new RecensioneServlet();
23
24         when(request.getParameter("utente")).thenReturn("Damian
25         when(request.getParameter("codicePacchetto")).thenReturn
26         when(request.getParameter("titoloRecensione")).thenReturn
27         when(request.getParameter("testoRecensione")).thenReturn
28
29         when(response.getWriter()).thenReturn(writer);
30         servlet.doGet(request, response);
31
32         String result = stringWriter.getBuffer().toString().tri
33         assertTrue(result.equals("{\"ok\":true,\"message\":\"\"}"));
34     }
35 }
```

The JUnit test results are shown in the right-hand pane. The test suite `Test_RecensioneServlet` (Runner: JUnit 5) finished after 2,008 seconds. It contains one test: `recensione()` (1,661 s). The test passed successfully.

5 Test di integrazione Manager

5.1 AcquisitoManager

The screenshot shows an IDE with two windows. The left window displays the `TestAcquisitoManager.java` file, which contains the following code:

```
1 package test.integrations.manager;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class TestAcquisitoManager {
6
7     @Test
8     void testGetDataOdierna() {
9         String data = manager.getDataOdierna();
10
11         assertNotNull(data);
12     }
13
14     @Test
15     void testGetOrdine() throws SQLException {
16         ArrayList<PacchettoBean> carrello = new ArrayList<PacchettoBean>();
17         PacchettoBean p = new PacchettoBean();
18         p.setCodicePacchetto("pac001");
19         p.setTitolo("Programmazione C");
20         p.setCategoria("Sviluppo");
21         p.setSottocategoria("svi001");
22         p.setPrezzo(98);
23         p.setDescrizione("Capire i fondamenti del linguaggio C");
24         p.setFoto("img/pacchetti/C1.jpg");
25         carrello.add(p);
26
27         manager.getOrdine("Damiana", carrello);
28
29         ArrayList<OrdineAcquistoBean> ordini = new ArrayList<OrdineAcquistoBean>();
30         OrdineManager ordine = new OrdineManager();
31         ordini = ordine.RicercaNomeCliente("Damiana");
32
33         assertNotNull(ordini);
34     }
35
36     AcquistoManager manager = new AcquistoManager();
37 }
```

The right window shows the JUnit test results for `TestAcquisitoManager`. The tests passed successfully:

- `testGetDataOdierna()` (0,043 s)
- `testGetOrdine()` (0,644 s)

The overall test run finished after 0,846 seconds with 2/2 runs, 0 errors, and 0 failures.

5.2 InsegnanteManager

The screenshot shows an IDE with two windows. The left window displays the `TestInsegnanteManager.java` file, which contains the following code:

```
1 package test.integrations.manager;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class TestInsegnanteManager {
6
7     @Test
8     void testInserisciPacchetto() {
9         PacchettoBean pacchettoInserito = new PacchettoBean();
10         pacchettoInserito = manager.inserPacchetto("pac052", "Marianosaria", "inf002", 53, "Routing e VLAN", "Co");
11
12         assertEquals("pac052", pacchettoInserito.getCodicePacchetto());
13     }
14
15     @Test
16     void testGetPacchetto() {
17         PacchettoBean pacchettoRestituito = new PacchettoBean();
18         pacchettoRestituito = manager.getPacchetto("pac004");
19
20         assertEquals("pac004", pacchettoRestituito.getCodicePacchetto());
21     }
22
23     @Test
24     void testGetPacchettoByTitolo() {
25         PacchettoBean pacchettoRestituito = new PacchettoBean();
26         pacchettoRestituito = manager.getPacchettoByTitolo("Programmazione C per principianti");
27
28         System.out.println(pacchettoRestituito.getCodicePacchetto());
29     }
30 }
```

The right window shows the JUnit test results for `TestInsegnanteManager`. The tests passed successfully:

- `testDeletePacchetto()` (0,743 s)
- `testGetPacchettoByTitolo()` (0,024 s)
- `testGetPacchetto()` (0,020 s)
- `testUpdatePrice()` (0,033 s)
- `testUpdateTitle()` (0,023 s)
- `testUpdateCode()` (0,019 s)
- `testUpdateDescription()` (0,029 s)
- `testDeleteLesson()` (0,047 s)
- `testGetAcquirenti()` (0,055 s)

The overall test run finished after 1,351 seconds with 16/16 runs, 0 errors, and 0 failures.

5.3 RecensioneManager

```
package test.integration.manager;

import static org.junit.Assert.assertNotNull;

class TestRecensioneManager {

    @Test
    void testAggiungiRecensione() {
        manager.aggiungiRecensione("Damiana", "pac043", "Pacchetto ottimo!", "Consigliatissimo!");

        ArrayList<RecensioneBean> recensioni = new ArrayList<RecensioneBean>();
        recensioni = managerLezioni.getRecensioni("pac043");
        RecensioneBean recensioneAttuale = null;

        for(RecensioneBean r : recensioni) {
            if((r.getCliente()).equals("Damiana"))
                recensioneAttuale = r;
        }

        assertNotNull(recensioneAttuale);
    }

    private RecensioneDao manager = new RecensioneDao();
    private LezioneManager managerLezioni = new LezioneManager();
}
```

Finished after 0,858 seconds

Runs: 1/1 Errors: 0 Failures: 0

TestRecensioneManager [Runner: JUnit 5] (0,653 s)

- testAggiungiRecensione() (0,653 s)

Failure Trace

5.4 SottocategoriaManager

```
package test.integration.manager;

import static org.junit.jupiter.api.Assertions.*;

class testSottocategoriaManager {

    @Test
    void testSelezionaSottocategorieInsegnante() {
        ArrayList<SottocategoriaBean> sottocategorie = new ArrayList<SottocategoriaBean>();
        sottocategorie = manager.selezionaSottocategorieInsegnante("Mariasaria");

        assertNotNull(sottocategorie);
    }

    @Test
    void testFindByKey() throws SQLException {
        SottocategoriaBean s = new SottocategoriaBean();
        s = (SottocategoriaBean) manager.findByKey("ins001");

        assertNotNull(s);
    }

    SottocategoriaManager manager = new SottocategoriaManager();
}
```

Finished after 0,845 seconds

Runs: 2/2 Errors: 0 Failures: 0

testSottocategoriaManager [Runner: JUnit 5] (0,668 s)

- testSelezionaSottocategorieInsegnante() (0,651 s)
- testFindByKey() (0,017 s)

5.5 UtenteManager

```
Test_UtenteManager.java

35 @Test
36 public void testLogin() {
37     UtenteBean utenteLoggato = new UtenteBean();
38
39     String passwordNuova = "chicca";
40     String passwordBase64format = Base64.getEncoder().encodeToString(passwordNuova.getBytes());
41     utenteLoggato = manager.getUtente("Mariasaria", passwordBase64format);
42     assertNotNull(utenteLoggato);
43 }
44
45 @Test
46 public void testModificaEmail() {
47     boolean res = manager.setEmail("mesposito@gmail.com", "chiccaesp98@gmail.com");
48     assertTrue(res);
49 }
50
51 @Test
52 public void testModificaPassword() {
53     String passwordNuova = "passMariasaria";
54     String passwordBase64format = Base64.getEncoder().encodeToString(passwordNuova.getBytes());
55     boolean res = manager.setPassword("mesposito@gmail.com", passwordBase64format);
56     assertTrue(res);
57 }
58
59 @Test
60 public void testRegistrazione() {
61     String password = "Maria Ausiliatrice";
62     String passwordBase64format = Base64.getEncoder().encodeToString(password.getBytes());
}
```

JUnit

Finished after 1,138 seconds

Runs: 10/10 Errors: 0 Failures: 0

Test_UtenteManager [Runner: JUnit 5] (0,995 s)

Failure Trace

6.Introduzione

Questa sezione rappresenta l'attività di esecuzione dei test case descritti all'interno del Test Specification. I test verranno eseguiti con **Selenium** e ad ogni esecuzione verrà affiliata una tabella informativa in cui mantenere le seguenti informazioni:

- 1. Nome del test case:** Corrispondente al test case del Test Specification Document.
- 2. Data e ora:** Quando il test è stato eseguito
- 3. Tester che ha eseguito il problema:** Soggetto creatore ed esecutore del test.
- 4. Output atteso:** Output che si dovrebbe ottenere dall'esecuzione del test.
- 5. Output del sistema:** Output che si ottiene dall'esecuzione del test.
- 6. Esito del test:** Corretto o sbagliato a secondo se l'output atteso e del sistema coincide o meno.
- 7. Numero prove:** Numero di prove prima che il test abbia esito corretto.
- 8. Categoria:** Valori della category partition interessati nell'esecuzione del test.

7. Gestione Utente

7.1 Login

Nome del test case:	TCS1
Data e ora:	16/01/2020 09:36
Tester:	Maria Rachele Tripodi
Output atteso:	Il login non va a buon fine poiché i dati inseriti non sono presenti nel database.
Output del sistema:	Il login non va a buon fine poiché i dati inseriti non sono presenti nel database.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Esistenza

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/AreaUtente.jsp

	Command	Target	Value
3	click	css=li:nth-child(1) > .navbar-item	
4	click	id=loginImage	
5	click	id=myName	
6	type	id=myName	Rachele
7	click	id=myPass	
8	type	id=myPass	ciao
9	click	css=.bottoneDefault:nth-child(6)	
10	assert text	css=#messageError	Username o password errati

Command

Target

Value

Description

Log

Reference

6. type on id=myName with value Rachele OK 09:36:47

7. click on id=myPass OK 09:36:47

8. type on id=myPass with value ciao OK 09:36:47

9. click on css=.bottoneDefault:nth-child(6) OK 09:36:47

10. assertText on css=#messageError with value Username o password errati OK 09:36:48

TCS1* completed successfully 09:36:48

Nome del test case:	TCS2
Data e ora:	16/01/2020 09:39
Tester:	Maria Rachele Tripodi
Output atteso:	Il login non va a buon fine poiché un campo è vuoto.
Output del sistema:	Il login non va a buon fine poiché un campo è vuoto.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
4	click	id=myName	
5	type	id=myName	lucio
6	click	css=.bottonDefault:nth-child(6)	
7	mouse down at	css=.sfondo-login:nth-child(3)	872,618
8	mouse move at	css=.sfondo-login:nth-child(3)	872,618
9	mouse up at	css=.sfondo-login:nth-child(3)	872,618
10	click	css=.sfondo-login:nth-child(3)	
11	assert text	id=messageError	Tutti i parametri devono essere passati

Command

Target

Value

Description

Log Reference

7. mouseDownAt on css=.sfondo-login:nth-child(3) with value 872,618 OK 09:39:24

8. mouseMoveAt on css=.sfondo-login:nth-child(3) with value 872,618 OK 09:39:24

9. mouseUpAt on css=.sfondo-login:nth-child(3) with value 872,618 OK 09:39:24

10. click on css=.sfondo-login:nth-child(3) OK 09:39:25

11. assertText on id=messageError with value Tutti i parametri devono essere passati OK 09:39:25

TCS2 completed successfully 09:39:25

Nome del test case:	TCS4
Data e ora:	15/01/2020 15:33
Tester:	Maria Rachele Tripodi
Output atteso:	Il login va a buon fine poiché i dati inseriti rispettano tutte le condizioni.
Output del sistema:	Il login va a buon fine poiché i dati inseriti rispettano tutte le condizioni.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato, Esistenza

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
3 click	css=li:nth-child(1) > .navbar-item	
4 click	id=loginImage	
5 click	id=myName	
6 type	id=myName	Rachele
7 click	id=myPass	
8 type	id=myPass	mangiona
9 click	css=botoneDefault:nth-child(6)	
10 assert element present	css=.cerchio	

Command

Target

Value

Description

Log Reference

6. type on id=myName with value Rachele OK 15:33:02

7. click on id=myPass OK 15:33:02

8. type on id=myPass with value mangiona OK 15:33:02

9. click on css=botoneDefault:nth-child(6) OK 15:33:03

10. assertElementPresent on css=.cerchio OK 15:33:03

TCS4* completed successfully 15:33:03

7.2 Registrazione

Nome del test case:	TCS5
Data e ora:	15/01/2020 15:22
Tester:	Maria Rachele Tripodi
Output atteso:	La registrazione non va a buon fine poiché il nome Utente è già esistente.
Output del sistema:	La registrazione non va a buon fine poiché il nome Utente è già esistente.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Esistenza

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
8	type	id=newEmail	lucioanna@gmail.com
9	click	id=regPass	
10	type	id=regPass	annafulgione
11	click	id=regPassRepeat	
12	type	id=regPassRepeat	annafulgione
13	click	css=.bottonDefault:nth-child(11)	
14	assert text	id=signUpMessageError	Utente già grave esistente

Command

Target

Value

Description

Log Reference

10. type on id=regPass with value annafulgione OK 15:22:23

11. click on id=regPassRepeat OK 15:22:23

12. type on id=regPassRepeat with value annafulgione OK 15:22:23

13. click on css=.bottonDefault:nth-child(11) OK 15:22:23

14. assertText on id=signUpMessageError with value Utente già grave esistente OK 15:22:23

TCS5* completed successfully 15:22:24

Nome del test case:	TCS6
Data e ora:	15/01/2020 15:42
Tester:	Maria Rachele Tripodi
Output atteso:	La registrazione non va a buon fine poiché il nome Utente non rispetta i vincoli di lunghezza.
Output del sistema:	La registrazione non va a buon fine poiché il nome Utente non rispetta i vincoli di lunghezza.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
11 type	id=regPass	francoforno
12 click	css=input-with-icon:nth-child(8) > .fas	
13 click	id=regPassRepeat	
14 type	id=regPassRepeat	francoforno
15 click	css=input-with-icon:nth-child(10) > .fas	
16 click	css=botoneDefault:nth-child(11)	
17 click	css=sfondo-login:nth-child(4)	
18 assert text	id=signUpMessageError	Inserire un nome utente da almeno 4 caratteri

Command

Target

Value

Description

Log Reference

14. type on id=regPassRepeat with value francoforno OK 15:42:54

15. click on css=input-with-icon:nth-child(10) > .fas OK 15:42:54

16. click on css=botoneDefault:nth-child(11) OK 15:42:55

17. click on css=sfondo-login:nth-child(4) OK 15:42:55

18. assertText on id=signUpMessageError with value Inserire un nome utente da almeno 4 caratteri OK 15:42:55

TCS6* completed successfully 15:42:55

Nome del test case:	TCS8
Data e ora:	15/01/2020 18:02
Tester:	Maria Rachele Tripodi
Output atteso:	La registrazione non va a buon fine poiché l'email non rispetta il vincolo di formato.
Output del sistema:	La registrazione non va a buon fine poiché l'email non rispetta il vincolo di formato.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
12 type	id=regPass	clara1999
13 click	css=input-with-icon:nth-child(8) > .fas	
14 click	id=regPassRepeat	
15 type	id=regPassRepeat	clara1999
16 click	css=input-with-icon:nth-child(10) > .fas	
17 click	css=botoneDefault:nth-child(11)	
18 assert text	id=signUpMessageError	Formato email non valido

Command

Target

Value

Description

Log Reference

14. click on id=regPassRepeat OK 18:02:54

15. type on id=regPassRepeat with value clara1999 OK 18:02:54

16. click on css=input-with-icon:nth-child(10) > .fas OK 18:02:54

17. click on css=botoneDefault:nth-child(11) OK 18:02:55

18. assertText on id=signUpMessageError with value Formato email non valido OK 18:02:55

'TCS8' completed successfully 18:02:55

Nome del test case:	TCS9
Data e ora:	15/01/2020 18:06
Tester:	Maria Rachele Tripodi
Output atteso:	La registrazione non va a buon fine poiché la password non rispetta il vincolo di lunghezza.
Output del sistema:	La registrazione non va a buon fine poiché la password non rispetta il vincolo di lunghezza.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
TCS3	7 click	id=newEmail	
TCS4	8 type	id=newEmail	carla@gmail.com
TCS5	9 click	id=regPass	
TCS6	10 type	id=regPass	car
TCS8	11 click	id=regPassRepeat	
TCS9*	12 type	id=regPassRepeat	car
TCS10	13 click	css=.bottonDefault:nth-child(11)	
TCS12	14 assert text	id=signUpMessageError	Inserire una password da almeno 8 caratteri

Command Target Value Description

Log Reference

10. type on id=regPass with value car OK 18:06:08

11. click on id=regPassRepeat OK 18:06:09

12. type on id=regPassRepeat with value car OK 18:06:09

13. click on css=.bottonDefault:nth-child(11) OK 18:06:09

14. assertText on id=signUpMessageError with value Inserire una password da almeno 8 caratteri OK 18:06:09

TCS9* completed successfully 18:06:09

Nome del test case:	TCS10
Data e ora:	15/01/2020 16:01
Tester:	Maria Rachele Tripodi
Output atteso:	La registrazione non va a buon fine poiché le due password non coincidono.
Output del sistema:	La registrazione non va a buon fine poiché le due password non coincidono.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
12 click	css=input-with-icon:nth-child(8) > .tas	
13 click	id=regPass	
14 type	id=regPass	annalucio
15 click	id=regPassRepeat	
16 type	id=regPassRepeat	Lucioanna
17 click	css=botoneDefault:nth-child(11)	
18 assert text	id=signUpMessageError	Le password non coincidono

Command

Target

Value

Description

Log Reference

14. type on id=regPass with value annalucio OK 16:01:54

15. click on id=regPassRepeat OK 16:01:54

16. type on id=regPassRepeat with value Lucioanna OK 16:01:55

17. click on css=botoneDefault:nth-child(11) OK 16:01:55

18. assertText on id=signUpMessageError with value Le password non coincidono OK 16:01:55

TCS10* completed successfully 16:01:55

Nome del test case:	TCS11
Data e ora:	16/01/2020 12:39
Tester:	Maria Rachele Tripodi
Output atteso:	La registrazione va a buon fine dato che tutti i vincoli sono stati rispettati.
Output del sistema:	La registrazione va a buon fine dato che tutti i vincoli sono stati rispettati.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMe

Project: StudyMe

Executing

TCS11

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
8	type	id=newEmail	anna@gmail.com
9	click	id=regPass	
10	type	id=regPass	fulgioneanna
11	click	id=regPassRepeat	
12	type	id=regPassRepeat	fulgioneanna
13	click	css= bottoneDefault:nth-child(11)	
14	click	css=.nav h1	
15	assert element present	css=li:nth-child(2) img	

Command

Target

Value

Description

Runs: 1 Failures: 0

Log	Reference
11. click on id=regPassRepeat OK	12:39:04
12. type on id=regPassRepeat with value fulgioneanna OK	12:39:05
13. click on css= bottoneDefault:nth-child(11) OK	12:39:05
14. click on css=.nav h1 OK	12:39:05
15. assertElementPresent on css=li:nth-child(2) img OK	12:39:05
TCS11 completed successfully	12:39:06

7.3 Cambio email

Nome del test case:	TCS12
Data e ora:	15/01/2020 17:57
Tester:	Maria Rachele Tripodi
Output atteso:	La modifica non va a buon fine perché la e-mail non rispetta il formato.
Output del sistema:	La modifica non va a buon fine perché la e-mail non rispetta il formato.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
TCS3	2	set window size	1382x744
TCS4	3	click	css=.nav h1
TCS5	4	click	id=changeEmail
TCS6	5	type	id=changeEmail a
TCS10	6	click	css=.bottoneDefault:nth-child(2)
TCS12*	7	assert text	id=messageErrorEmail Formato email non valido
TCS14	Command		
TCS15	Target		
TCS16	Value		
TCS17	Description		

Log Reference

3. click on css=.nav h1 OK 17:57:40

4. click on id=changeEmail OK 17:57:40

5. type on id=changeEmail with value a OK 17:57:41

6. click on css=.bottoneDefault:nth-child(2) OK 17:57:41

7. assertText on id=messageErrorEmail with value Formato email non valido OK 17:57:41

TCS12 completed successfully 17:57:41

Nome del test case:	TCS13
Data e ora:	16/01/2020
Tester:	Maria Rachele Tripodi
Output atteso:	La modifica non va a buon fine perché la e-mail è già presente nel database di sistema.
Output del sistema:	La modifica non va a buon fine perché la e-mail è già presente nel database di sistema.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Esistenza


Selenium IDE - StudyMe



Project: StudyMe

Tests +

Search tests...

	Command	Target	Value
TCS3	1 open	http://localhost:8080/Source/AreaUtente.jsp	
TCS4	2 set window size	1366x728	
TCS5	3 click	id=changeEmail	
TCS6	4 type	id=changeEmail	lorenzo@live.it
TCS8	5 click	css=bottoneDefault:nth-child(2)	
TCS9	6 assert text	id=messageErrorEmail	L'email è già presente nel sistema
TCS10			
TCS12			
TCS13			
TCS14			
TCS15			
TCS16			
TCS17			

Command # 

Target  

Value

Description

Log Reference

2. setWindowSize on 1366x728 OK 09:16:32

3. click on id=changeEmail OK 09:16:32

4. type on id=changeEmail with value lorenzo@live.it OK 09:16:33

5. click on css=bottoneDefault:nth-child(2) OK 09:16:33

6. assertText on id=messageErrorEmail with value L'email è già presente nel sistema OK 09:16:33

'TCS13' completed successfully 09:16:33

Nome del test case:	TCS14
Data e ora:	15/01/2020 16:20
Tester:	Maria Rachele Tripodi
Output atteso:	La modifica va a buon fine poiché vengono rispettate tutte le condizioni.
Output del sistema:	La modifica va a buon fine poiché vengono rispettate tutte le condizioni.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato, Esistenza

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
1	open	http://localhost:8080/Source/HomePage.jsp	
2	set window size	1382x744	
3	click	css= nav h1	
4	click	id=changeEmail	
5	type	id=changeEmail	anna@gmail.com
6	click	css= bottoneDefault:nth-child(2)	
7	assert text	id=email	Email modificata

Command

Target

Value

Description

Log Reference

3. click on css= nav h1 OK 16:20:00

4. click on id=changeEmail OK 16:20:01

5. type on id=changeEmail with value anna@gmail.com OK 16:20:01

6. click on css= bottoneDefault:nth-child(2) OK 16:20:01

7. assertText on id=email with value Email modificata OK 16:20:02

TCS14* completed successfully 16:20:02

7.4 Cambio password

Nome del test case:	TCS15
Data e ora:	15/01/2020 16:24
Tester:	Maria Rachele Tripodi
Output atteso:	La modifica non va a buon fine poiché i campi non sono uguali.
Output del sistema:	La modifica non va a buon fine poiché i campi non sono uguali.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
2 set window size	1382x744	
3 click	css= nav h1	
4 click	id=changePassword	
5 type	id=changePassword	annalucio
6 click	id=confChangePassword	
7 type	id=confChangePassword	annafulgione
8 click	css=#formPassword > .bottoneDefault	
9 assert text	id=messageErrorPassword	Le password non coincidono

Command

Target

Value

Description

Log Reference

5. type on id=changePassword with value annalucio OK 16:24:02

6. click on id=confChangePassword OK 16:24:03

7. type on id=confChangePassword with value annafulgione OK 16:24:03

8. click on css=#formPassword > .bottoneDefault OK 16:24:03

9. assertText on id=messageErrorPassword with value Le password non coincidono OK 16:24:03

TCS15* completed successfully 16:24:04

Nome del test case:	TCS16
Data e ora:	15/01/2020 17:55
Tester:	Maria Rachele Tripodi
Output atteso:	La modifica non va a buon fine poiché i campi non rispettano la lunghezza della password.
Output del sistema:	La modifica non va a buon fine poiché i campi non rispettano la lunghezza della password.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
TCS3	set window size	1382x744	
TCS4	click	css=nav h1	
TCS5	click	id=changePassword	
TCS6	type	id=changePassword	a
TCS10	click	id=confChangePassword	
TCS14	type	id=confChangePassword	a
TCS15	click	css=#formPassword > .botoneDefault	
TCS16*	assert text	id=messageErrorPassword	Inserire password di almeno 8 caratteri

TCS17

Command

Target

Value

Description

Log Reference

5. type on id=changePassword with value a OK 17:55:55

6. click on id=confChangePassword OK 17:55:55

7. type on id=confChangePassword with value a OK 17:55:55

8. click on css=#formPassword > .botoneDefault OK 17:55:56

9. assertText on id=messageErrorPassword with value Inserire password di almeno 8 caratteri OK 17:55:56

'TCS16' completed successfully 17:55:56

Nome del test case:	TCS17
Data e ora:	15/01/2020 16:27
Tester:	Maria Rachele Tripodi
Output atteso:	La modifica va a buon fine poiché tutte le condizioni sono verificate.
Output del sistema:	La modifica va a buon fine poiché tutte le condizioni sono verificate.
Esito test:	CORRETTO
Numero prove:	1
Categoria:	Formato, Esistenza

Selenium IDE - StudyMe*

Project: StudyMe*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
12 click	id=confChangePassword	
13 type	id=confChangePassword	annalucio
14 click	css=input-with-icon:nth-child(3) > .fas	
15 click	id=confChangePassword	
16 type	id=confChangePassword	annalucio
17 click	css=#formPassword > .bottoneDefault	
18 assert text	id=password	Password modificata

TCS3

TCS4

TCS5

TCS6

TCS10*

TCS14*

TCS15*

TCS17*

Command

Target

Value

Description

Log Reference

14. click on css=input-with-icon:nth-child(3) > .fas OK 16:27:21

15. click on id=confChangePassword OK 16:27:21

16. type on id=confChangePassword with value annalucio OK 16:27:21

17. click on css=#formPassword > .bottoneDefault OK 16:27:22

18. assertText on id=password with value Password modificata OK 16:27:22

TCS17* completed successfully 16:27:22

8. Gestione pacchetto

8.1 Crea Pacchetto

Nome del test case:	TCS18
Data e ora:	16 Gennaio 2020 9:40
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché il codice del pacchetto è già esistente.
Output del sistema:	La creazione non va a buon fine perché il codice del pacchetto è già esistente.
Esito del test:	Corretto
Numero prove:	1
Categoria:	EsistenzaC

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/StudyMe/AreaUtente.jsp

	Command	Target	Value
TCS18*	13 click	id=newPrice	
TCS20	14 type	id=newPrice	35
TCS21	15 click	id=newDesc	
TCS22	16 type	id=newDesc	Introduzione corso reti
TCS23	17 click	css= bottoneDefault:nth-child(15)	
TCS24	18 assert text	id=success	Codice pacchetto già in uso
TCS25			
TCS26	Command		
TCS27	Target		
TCS28*	Value		
TCS29*	Description		
TCS30*			

Log Reference

14. type on id=newPrice with value 35 OK 09:40:48

15. click on id=newDesc OK 09:40:48

16. type on id=newDesc with value Introduzione corso reti OK 09:40:49

17. click on css= bottoneDefault:nth-child(15) OK 09:40:49

18. assertText on id=success with value Codice pacchetto già in uso OK 09:40:49

TCS18* completed successfully 09:40:49

Nome del test case:	TCS20
Data e ora:	15 Gennaio 2020 16:50
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché non rispetta i criteri di lunghezza per il titolo.
Output del sistema:	La creazione non va a buon fine perché non rispetta i criteri di lunghezza per il titolo.
Esito del test:	Corretto
Numero prove:	1
Categoria:	LunghezzaT

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/StudyMe/HomePage.jsp

	Command	Target	Value
TCS20*			
TCS33	15 click	id=newPrice	
TCS34	16 type	id=newPrice	55
TCS35	17 click	id=newDesc	
TCS36	18 click	id=newDesc	
TCS37	19 type	id=newDesc	Imparare a conoscere i threads
TCS38	20 click	css= bottoneDefault:nth-child(15)	
	21 assert text	id=success	Inserire un titolo compreso tra i 5 e 35 caratteri

Command

Target

Value

Description

Log Reference

17. click on id=newDesc OK 16:50:23

18. click on id=newDesc OK 16:50:24

19. type on id=newDesc with value Imparare a conoscere i threads OK 16:50:24

20. click on css= bottoneDefault:nth-child(15) OK 16:50:24

21. assertText on id=success with value Inserire un titolo compreso tra i 5 e 35 caratteri OK 16:50:24

TCS20* completed successfully 16:50:25

Nome del test case:	TCS21
Data e ora:	15 Gennaio 2020 17:02
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché il pacchetto con il titolo "Imparare a conoscere i threads" è già esistente.
Output del sistema:	La creazione non va a buon fine perché il pacchetto con il titolo "Imparare a conoscere i threads" è già esistente.
Esito del test:	Corretto
Numero prove:	1
Categoria:	EsistenzaT

Project: StudyMeSelenium*

Tests + [Icons]

Search tests... [Icon]

http://localhost:8080/StudyMe/HomePage.jsp

	Command	Target	Value
TCS20*	13 type	id=newPhoto	https://www.sinonimi-contrari.it/includes/images/sharers/sinonimi-di-prova.png
TCS21*	14 click	id=newPrice	
TCS33	15 type	id=newPrice	55
TCS34	16 click	id=newDesc	
TCS35	17 type	id=newDesc	Imparare a programmare in shell
TCS36	18 click	css=.bottoneDefault:nth-child(15)	
TCS37	19 assert text	id=success	Titolo già esistente
TCS38	Command	[Icon]	
	Target	[Icon]	
	Value		
	Description		

Log Reference

15. type on id=newPrice with value 55 OK 17:02:56

16. click on id=newDesc OK 17:02:56

17. type on id=newDesc with value Imparare a programmare in shell OK 17:02:57

18. click on css=.bottoneDefault:nth-child(15) OK 17:02:57

19. assertText on id=success with value Titolo già esistente OK 17:02:57

'TCS21' completed successfully 17:02:58

Nome del test case:	TCS22
Data e ora:	15 Gennaio 2020 17:34
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché il Prezzo non rispetta i criteri di formato.
Output del sistema:	La creazione non va a buon fine perché il Prezzo non rispetta i criteri di formato.
Esito del test:	Corretto
Numero prove:	1
Categoria:	FormatoP

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/StudyMe/HomePage.jsp

	Command	Target	Value
TCS20*	16 type	id=newPhoto	https://www.sinonimi-contrari.it/includes/images/sharers/sinonimi-di-prova.png
TCS21*	17 click	id=newPrice	
TCS22*	18 type	id=newPrice	AA
TCS33	19 click	id=newDesc	
TCS34	20 type	id=newDesc	Imparare ad usare i threads
TCS35	21 click	css=.bottoneDefault:nth-child(15)	
TCS36	22 assert text	id=success	Prezzo non valido

Command

Target

Value

Description

Log Reference

18. type on id=newPrice with value AA OK 17:34:04

19. click on id=newDesc OK 17:34:04

20. type on id=newDesc with value Imparare ad usare i threads OK 17:34:04

21. click on css=.bottoneDefault:nth-child(15) OK 17:34:05

22. assertText on id=success with value Prezzo non valido OK 17:34:05

TCS22' completed successfully 17:34:05

Nome del test case:	TCS23
Data e ora:	15 Gennaio 2020 17:44
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché non è stato compilato il campo di Descrizione.
Output del sistema:	La creazione non va a buon fine perché non è stato compilato il campo di Descrizione.
Esito del test:	Corretto
Numero prove:	1
Categoria:	EsistenzaD

Project: StudyMeSelenium*

Tests ▾ + ▶ ⏮ ⏭ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷

Search tests... 🔍

http://localhost:8080/StudyMe/HomePage.jsp

	Command	Target	Value
TCS20*	11 type	id=newTitle	Imparare ad usare i threads
TCS21*	12 click	id=newPhoto	
TCS22*	13 type	id=newPhoto	https://www.sinonimi-contrari.it/includes/images/sharers/sinonimi-di-prova.png
TCS23*	14 click	id=newPrice	
TCS33	15 type	id=newPrice	55
TCS34	16 click	css=.bottoneDefault:nth-child(15)	
TCS35	17 assert text	id=success	Tutti i parametri devono essere compilati

Command #
 Target
 Value
 Description

Log Reference

13. type on id=newPhoto with value https://www.sinonimi-contrari.it/includes/images/sharers/sinonimi-di-prova.png OK 17:44:56

14. click on id=newPrice OK 17:44:56

15. type on id=newPrice with value 55 OK 17:44:56

16. click on css=.bottoneDefault:nth-child(15) OK 17:44:56

17. assertText on id=success with value Tutti i parametri devono essere compilati OK 17:44:57

***TCS23* completed successfully** 17:44:57

Nome del test case:	TCS24
Data e ora:	15 Gennaio 2020 17:52
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché la Descrizione non rispetta i criteri di lunghezza.
Output del sistema:	La creazione non va a buon fine perché la Descrizione non rispetta i criteri di lunghezza.
Esito del test:	Corretto
Numero prove:	1
Categoria:	LunghezzaD

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/StudyMe/HomePage.jsp

	Command	Target	Value
TCS20	12 type	id=newPhoto	https://www.sinonimi-contrari.it/includes/images/sharers/sinonimi-di-prova.png
TCS21	13 click	id=newPrice	
TCS22	14 type	id=newPrice	55
TCS23	15 click	id=newDesc	
TCS24*	16 type	id=newDesc	A
TCS33	17 click	css=.bottoneDefault:nth-child(15)	
TCS34	18 assert text	id=success	Inserire una descrizione compresa tra i 10 e i 30 caratteri

Command

Target

Value

Description

Log Reference

14. type on id=newPrice with value 55 OK 17:52:17

15. click on id=newDesc OK 17:52:18

16. type on id=newDesc with value A OK 17:52:18

17. click on css=.bottoneDefault:nth-child(15) OK 17:52:18

18. assertText on id=success with value Inserire una descrizione compresa tra i 10 e i 30 caratteri OK 17:52:19

TCS24 completed successfully 17:52:19

Nome del test case:	TCS25
Data e ora:	16 Gennaio 2020 12:03
Tester:	Mariarosaria Esposito
Output atteso:	L'insegnante può proseguire con la creazione perchè tutti i campi sono stati compilati correttamente.
Output del sistema:	L'insegnante può proseguire con la creazione perchè tutti i campi sono stati compilati correttamente.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Formato, Esistenza, Lunghezza

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/StudyMe/AreaUtente.jsp

	Command	Target	Value
TCS18			
TCS20	12 type	id=newPhoto	https://www.serin.pa.it/images/servizi_reti.jpg
TCS21	13 click	id=newPrice	
TCS22	14 type	id=newPrice	45
TCS23	15 click	id=newDesc	
TCS24	16 type	id=newDesc	Corso di reti
TCS25*	17 click	css= .bottoneDefault:nth-child(15)	
	18 assert element present	id=lezioni	
TCS26	Command		
TCS27	Target		
TCS28	Value		
TCS29	Description		
TCS30			

Log Reference

14. type on id=newPrice with value 45 OK 12:03:22

15. click on id=newDesc OK 12:03:22

16. type on id=newDesc with value Corso di reti OK 12:03:22

17. click on css= .bottoneDefault:nth-child(15) OK 12:03:22

18. assertElementPresent on id=lezioni OK 12:03:23

TCS25* completed successfully 12:03:23

Nome del test case:	TCS26
Data e ora:	15 Gennaio 2020 18:31
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché l'URL non rispetta i criteri di formato.
Output del sistema:	La creazione non va a buon fine perché l'URL non rispetta i criteri di formato.
Esito del test:	Corretto
Numero prove:	1
Categoria:	FormatoU

Project: StudyMeSelenium*

Executing

TCS26*

http://localhost:8080/StudyMe/AreaUtente.jsp

Command	Target	Value
17 click	css= bottoneDefault:nth-child(15)	
18 type	id=url	AA
19 type	id=title	Corso sistemi 2
20 type	id=duration	45:00
21 click	css= bottoneDefault:nth-child(9)	
22 assert text	id=messErr	Url non valido!

Command

Target

Value

Description

Runs: 1 Failures: 0

Log Reference

18. type on id=url with value AA OK 18:31:06

19. type on id=title with value Corso sistemi 2 OK 18:31:07

20. type on id=duration with value 45:00 OK 18:31:07

21. click on css= bottoneDefault:nth-child(9) OK 18:31:07

22. assertText on id=messErr with value Url non valido! OK 18:31:07

TCS26* completed successfully 18:31:08

Nome del test case:	TCS28
Data e ora:	16 Gennaio 2020 8:50
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché il Titolo non rispetta i criteri di lunghezza.
Output del sistema:	La creazione non va a buon fine perché il Titolo non rispetta i criteri di lunghezza.
Esito del test:	Corretto
Numero prove:	1
Categoria:	LunghezzaT

Project: StudyMeSelenium*

Tests +

Search tests... <http://localhost:8080/StudyMe/AreaUtente.jsp>

	Command	Target	Value
TCS20			
TCS21	19 type	id=url	https://www.youtube.com/embed/QRMTBL4u-8
TCS22	20 click	id=title	
TCS23	21 type	id=title	AA
TCS24	22 click	id=duration	
TCS25	23 type	id=duration	55:00
TCS26	24 click	css=botoneDefault:nth-child(9)	
TCS27	25 assert text	id=messErr	Inserire un titolo compreso tra i 5 e 35 caratteri
TCS28*	Command		
TCS33	Target		
TCS34	Value		
TCS35	Description		

Log Reference

21. type on id=title with value AA OK 08:50:56

22. click on id=duration OK 08:50:56

23. type on id=duration with value 55:00 OK 08:50:56

24. click on css=botoneDefault:nth-child(9) OK 08:50:56

25. assertText on id=messErr with value Inserire un titolo compreso tra i 5 e 35 caratteri OK 08:50:57

***TCS28* completed successfully** 08:50:57

Nome del test case:	TCS29
Data e ora:	16 Gennaio 2020 9:02
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché esiste già una lezione con il titolo "IMPARARE LA SHELL - INTRODUZIONE".
Output del sistema:	La creazione non va a buon fine perché esiste già una lezione con il titolo "IMPARARE LA SHELL - INTRODUZIONE".
Esito del test:	Corretto
Numero prove:	1
Categoria:	EsistenzaT

Project: StudyMeSelenium*

Tests + [Icons]

Search tests... [Icon] http://localhost:8080/StudyMe/AreaUtente.jsp [Icon]

	Command	Target	Value
TCS20			
TCS21	20 type	id=url	https://www.youtube.com/embed/Hae3MFGs2Ww
TCS22	21 click	id=title	
TCS23	22 type	id=title	IMPARARE LA SHELL - INTRODUZIONE
TCS24	23 click	id=duration	
TCS25	24 type	id=duration	55:00
TCS26	25 click	css=.bottonDefault:nth-child(9)	
TCS27	26 assert text	id=messErr	Titolo già esistente
TCS28*	Command	[Icon]	[Icon]
TCS29*	Target	[Icon]	[Icon]
TCS30	Value		
TCS31	Description		
TCS32			
TCS33			
TCS34			

Log Reference

22. type on id=title with value IMPARARE LA SHELL - INTRODUZIONE OK 09:02:04

23. click on id=duration OK 09:02:04

24. type on id=duration with value 55:00 OK 09:02:04

25. click on css=.bottonDefault:nth-child(9) OK 09:02:05

26. assertText on id=messErr with value Titolo già esistente OK 09:02:05

TCS29 completed successfully 09:02:05

Nome del test case:	TCS30
Data e ora:	16 Gennaio 2020 9:07
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché non tutti i campi sono stati compilati.
Output del sistema:	La creazione non va a buon fine perché non tutti i campi sono stati compilati.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Formato

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/StudyMe/AreaUtente.jsp

	Command	Target	Value
TCS20			
TCS21	17 click	css= bottoneDefault:nth-child(15)	
TCS22	18 click	id=url	
TCS23	19 type	id=url	https://www.youtube.com/embed/u-sC7b0WmgQ
TCS24	20 click	id=title	
TCS25	21 type	id=title	Lezione 1
TCS26	22 click	css= bottoneDefault:nth-child(9)	
TCS27	23 assert text	id=messErr	Tutti i parametri devono essere compilati
TCS28*	Command		
TCS29*	Target		
TCS30*	Value		
TCS33	Description		

Log Reference

19. type on id=url with value https://www.youtube.com/embed/u-sC7b0WmgQ OK 09:07:27

20. click on id=title OK 09:07:27

21. type on id=title with value Lezione 1 OK 09:07:27

22. click on css= bottoneDefault:nth-child(9) OK 09:07:27

23. assertText on id=messErr with value Tutti i parametri devono essere compilati OK 09:07:28

TCS30* completed successfully 09:07:28

Nome del test case:	TCS32
Data e ora:	16 Gennaio 2020 12:19
Tester:	Mariarosaria Esposito
Output atteso:	L'insegnante può proseguire con la creazione perché tutti i campi sono stati compilati correttamente.
Output del sistema:	L'insegnante può proseguire con la creazione perché tutti i campi sono stati compilati correttamente.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Formato, Esistenza, Lunghezza

Project: StudyMeSelenium*

Executing

TCS32*

http://localhost:8080/StudyMe/AreaUtente.jsp

	Command	Target	Value
21	type	id=url	https://www.youtube.com/embed_PgVmkfBm14
22	click	id=title	
23	type	id=title	Lezione 1 di 3
24	click	id=duration	
25	type	id=duration	18:00
26	click	css=.bottoneDefault:nth-child(9)	
27	assert element present	id=riepilogo	

Runs: 1 Failures: 0

Log Reference

23. type on id=title with value Lezione 1 di 3 OK 12:19:06

24. click on id=duration OK 12:19:06

25. type on id=duration with value 18:00 OK 12:19:06

26. click on css=.bottoneDefault:nth-child(9) OK 12:19:06

27. assertElementPresent on id=riepilogo OK 12:19:06

TCS32* completed successfully 12:19:06

8.2 Modifica pacchetto

Nome del test case:	TCS33
Data e ora:	15 Gennaio 2020 15:31
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché il Codice Pacchetto è già esistente.
Output del sistema:	La creazione non va a buon fine perché il Codice Pacchetto è già esistente.
Esito del test:	Corretto
Numero prove:	1
Categoria:	EsistenzaP

Project: StudyMeSelenium

Tests +

Search tests...

http://localhost:8080/StudyMe/HomePage.jsp

	Command	Target	Value
2	set window size	1364x738	
3	click	css=a:nth-child(4).sfondo	
4	click	css=sottocat:nth-child(8).modifica	
5	click	id=newCode	
6	type	id=newCode	pac001
7	click	id=updateButton	
8	assert text	id=insuccess	Codice pacchetto già in uso

Command

Target

Value

Description

Log Reference

4. click on css=sottocat:nth-child(8).modifica OK 15:31:40

5. click on id=newCode OK 15:31:40

6. type on id=newCode with value pac001 OK 15:31:40






7. click on id=updateButton OK 15:31:41


8. assertText on id=insuccess with value Codice pacchetto già in uso OK 15:31:41


TCS33' completed successfully 15:31:41

Nome del test case:	TCS34
Data e ora:	15 Gennaio 2020 15:19
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché il Titolo non rispetta i criteri di lunghezza.
Output del sistema:	La creazione non va a buon fine perchè il Titolo non rispetta i criteri di lunghezza.
Esito del test:	Corretto
Numero prove:	1
Categoria:	LunghezzaT




Project: StudyMeSelenium



Tests +     

Search tests... 

http://localhost:8080/StudyMe/HomePage.jsp 

	Command	Target	Value
TCS34			
TCS35			
2	set window size	1364x738	
3	click	css=a:nth-child(4).sfondo	
4	click	css=sottocat:nth-child(8).modifica	
5	click	id=newTitle	
6	type	id=newTitle	A
7	click	css=#updateTitle > #updateButton	
8	assert text	id=insuccess	Inserire un titolo compreso tra i 5 e 35 caratteri

Command   

Target  

Value

Description

Log Reference

4. click on css=sottocat:nth-child(8).modifica OK 15:19:19

5. click on id=newTitle OK 15:19:20

6. type on id=newTitle with value A OK 15:19:20

7. click on css=#updateTitle > #updateButton OK 15:19:20

8. assertText on id=insuccess with value Inserire un titolo compreso tra i 5 e 35 caratteri OK 15:19:20

'TCS34' completed successfully 15:19:20

Nome del test case:	TCS35
Data e ora:	15 Gennaio 2020 15:17
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché il Titolo è già assegnato ad un altro pacchetto.
Output del sistema:	La creazione non va a buon fine perchè il Titolo è già assegnato ad un altro pacchetto.
Esito del test:	Corretto
Numero prove:	1
Categoria:	EsistenzaT

Project: StudyMeSelenium*

Tests +

Search tests...

URL: http://localhost:8080/StudyMe/HomePage.jsp

Command	Target	Value
2 set window size	1364x738	
3 click	css=a:nth-child(4).sfondo	
4 click	css=sottocat:nth-child(8).modifica	
5 click	id=newTitle	
6 type	id=newTitle	Programmazione della shell Linux
7 click	css=#updateTitle > #updateButton	
8 assert text	id=insuccess	Titolo già esistente

Command #

Target

Value

Description

Log Reference

- 4. click on css=sottocat:nth-child(8).modifica OK 15:17:34
- 5. click on id=newTitle OK 15:17:34
- 6. type on id=newTitle with value Programmazione della shell Linux OK 15:17:35
- 7. click on css=#updateTitle > #updateButton OK 15:17:35
- 8. assertText on id=insuccess with value Titolo già esistente OK 15:17:35

'TCS35' completed successfully 15:17:35

Nome del test case:	TCS36
Data e ora:	15 Gennaio 2020 15:22
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché il Prezzo non rispetta i criteri di formato.
Output del sistema:	La creazione non va a buon fine perchè il Prezzo non rispetta i criteri di formato.
Esito del test:	Corretto
Numero prove:	1
Categoria:	FormatoP

Project: StudyMeSelenium*

Tests +

Search tests...

TCS34

TCS35

TCS36*

http://localhost:8080/StudyMe/HomePage.jsp

	Command	Target	Value
2	set window size	1364x738	
3	click	css=a:nth-child(4).sfondo	
4	click	css=sottocat:nth-child(8).modifica	
5	click	id=newPrice	
6	type	id=newPrice	A
7	click	css=#updatePrice > #updateButton	
8	assert text	id=insuccess	Prezzo non valido

Command

Target

Value

Description

Log Reference

4. click on css=sottocat:nth-child(8).modifica OK 15:22:08

5. click on id=newPrice OK 15:22:09

6. type on id=newPrice with value A OK 15:22:09






7. click on css=#updatePrice > #updateButton OK 15:22:09


8. assertText on id=insuccess with value Prezzo non valido OK 15:22:10

TCS36* completed successfully 15:22:10

Nome del test case:	TCS37
Data e ora:	15 Gennaio 2020 15:24
Tester:	Mariarosaria Esposito
Output atteso:	La creazione non va a buon fine perché la Descrizione non rispetta i criteri di lunghezza.
Output del sistema:	La creazione non va a buon fine perché la Descrizione non rispetta i criteri di lunghezza.
Esito del test:	Corretto
Numero prove:	1
Categoria:	LunghezzaD

Project: StudyMeSelenium



Tests +     



Search tests... 

TCS34
TCS35
TCS36
TCS37

URL: <http://localhost:8080/StudyMe/HomePage.jsp>

	Command	Target	Value
2	set window size	1364x738	
3	click	css=a:nth-child(4).sfondo	
4	click	css=sottocat:nth-child(8).modifica	
5	click	id=newDesc	
6	type	id=newDesc	A
7	click	css=#updateDescr > #updateButton	
8	assert text	id=insuccess	Inserire una descrizione compresa tra i 10 e i 30 caratteri

Command:  

Target:  

Value:

Description:

Log Reference

4. click on css=sottocat:nth-child(8).modifica OK	15:24:37
5. click on id=newDesc OK	15:24:37
6. type on id=newDesc with value A OK	15:24:38
7. click on css=#updateDescr > #updateButton OK	15:24:38
8. assertText on id=insuccess with value Inserire una descrizione compresa tra i 10 e i 30 caratteri OK	15:24:38
TCS37 completed successfully	15:24:38

Nome del test case:	TCS38
Data e ora:	15 Gennaio 2020 15:24
Tester:	Mariarosaria Esposito
Output atteso:	La modifica va a buon fine in quanto il campo è stato inserito correttamente.
Output del sistema:	La modifica va a buon fine in quanto il campo è stato inserito correttamente.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Formato, Esistenza, Lunghezza

Project: StudyMeSelenium

Tests + ▶ ⌂ 🔍 🔄

Search tests... 🔍 http://localhost:8080/StudyMe/AreaUtente.jsp

Command	Target	Value
2 <i>set window size</i>	1260x718	
3 <i>click</i>	css=a.nth-child(4).sfondo	
4 <i>click</i>	css=sottocat:nth-child(8).pacchetto:nth-child(1).modifica	
5 <i>click</i>	id=newTitle	
6 <i>type</i>	id=newTitle	Shell Linux
7 <i>click</i>	css=#updateTitle > #updateButton	
8 <i>assert element present</i>	css=sottocat:nth-child(8).pacchetto:nth-child(1) > h1	Shell Linux

Command ⌵ # 🔍

Target 🔍 🔍

Value

Description

Log Reference

4. click on css=sottocat:nth-child(8).pacchetto:nth-child(1).modifica OK 12:25:40

5. click on id=newTitle OK 12:25:41

6. type on id=newTitle with value Shell Linux OK 12:25:41

7. click on css=#updateTitle > #updateButton OK 12:25:41

8. assertElementPresent on css=sottocat:nth-child(8).pacchetto:nth-child(1) > h1 with value Shell Linux OK 12:25:41

TCS38' completed successfully 12:25:42

9. Gestione Lezioni

9.1 Inserisci lezione

Nome test case:	TCS39
Data e ora:	15/01/2015:01
Tester:	Claudia Buono
Output atteso:	L'inserimento non va a buon fine in quanto l'URL non rispetta il criterio di formato.
Output reale:	L'inserimento non va a buon fine in quanto l'URL non rispetta il criterio di formato.
Esito del test:	Corretto
Numero prova:	1
Categoria:	Formato

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Executing

TCS39*

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
11	click	id=title	
12	type	id=title	prova
13	type	id=title	prova lezione nuova
14	type	id=duration	11:11
15	click	id=addLesson	
16	assertText	id=messErr	Url non valido!

Command:

Target:

Value:

Description:

Runs: 1 Failures: 0

Log Reference

11. click on id=title OK 15:01:27

12. type on id=title with value prova OK 15:01:27

13. type on id=title with value prova lezione nuova OK 15:01:28

14. type on id=duration with value 11:11 OK 15:01:28

15. click on id=addLesson OK 15:01:28

16. assertText on id=messErr with value Url non valido! OK 15:01:28

TCS39 completed successfully 15:01:28

Nome test case:	TCS40
Data e ora:	15/01/20 15:16
Tester:	Claudia Buono
Output atteso:	L'inserimento non va a buon fine in quanto la lezione con il seguente URL è già presente nel pacchetto.
Output reale:	L'inserimento non va a buon fine in quanto la lezione con il seguente URL è già presente nel pacchetto.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Esistenza

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

TCS39

TCS40*

Untitled

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
12. type	id=title	prova matematica
13. click	id=duration	
14. type	id=duration	11:11
15. click	id=addLesson	
16. assertText	id=messErr	Url già esistente

Command

Target

Value

Description

Log Reference

12. type on id=title with value prova matematica OK 15:16:31

13. click on id=duration OK 15:16:31

14. type on id=duration with value 11:11 OK 15:16:31

15. click on id=addLesson OK 15:16:31

16. assertText on id=messErr with value Url già esistente OK 15:16:31

TCS40 completed successfully 15:16:32

Nome test case:	TCS41
Data e ora:	15/01/20 15:35
Tester:	Claudia Buono
Output atteso:	L'inserimento non va a buon fine in quanto il Titolo non rispetta i criteri di lunghezza.
Output reale:	L'inserimento non va a buon fine in quanto il Titolo non rispetta i criteri di lunghezza.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Lunghezza

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
10 type	id=url	https://www.youtube.com/embed/KYjwcntuEkk
11 click	id=title	
12 type	id=title	A
13 click	id=duration	
14 type	id=duration	11:11
15 click	id=addLesson	
16 assert text	id=messErr	Inserire un titolo compreso tra i 5 e 35 caratteri

Command

Target

Value

Description

Log Reference

12. type on id=title with value A OK 15:35:40

13. click on id=duration OK 15:35:40

14. type on id=duration with value 11:11 OK 15:35:40

15. click on id=addLesson OK 15:35:40

16. assertText on id=messErr with value Inserire un titolo compreso tra i 5 e 35 caratteri OK 15:35:40

***TCS41* completed successfully** 15:35:40

Nome test case:	TCS42
Data e ora:	15/01/20 15:47
Tester:	Claudia Buono
Output atteso:	L'inserimento non va a buon fine in quanto non sono stati compilati tutti i campi.
Output reale:	L'inserimento non va a buon fine in quanto non sono stati compilati tutti i campi.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests: +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
8 click	id=url	
9 type	id=url	https://www.youtube.com/embed/KYjwntuEiKk
10 click	id=title	
11 click	id=title	
12 type	id=title	matematica nuova
13 click	id=addLesson	
14 assert text	id=messErr	Tutti i parametri devono essere compilati

Command: #

Target:

Value:

Description:

Log Reference

12. type on id=title with value matematica nuova OK 15:47:06

13. click on id=addLesson OK 15:47:06

14. assertText on id=messErr with value Tutti i parametri devono essere compilati OK 15:47:07

TCS42* completed successfully 15:47:07

Nome test case:	TCS43
Data e ora:	15/01/20 15:58
Tester:	Claudia Buono
Output atteso:	L'inserimento non va a buon fine in quanto nel pacchetto è già presente una lezione con il titolo "Lezione 1 – Violino".
Output reale:	L'inserimento non va a buon fine in quanto nel pacchetto è già presente una lezione con il titolo "Lezione 1 – Violino".
Esito del test:	Corretto
Numero prove:	1
Categoria:	Esistenza

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

TCS39

TCS40

TCS41

TCS42

TCS43*

Untitled

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
12. type	id=url	https://www.youtube.com/embed/JHp8DNQrVHs
13. click	id=title	
14. type	id=title	Lezione 1 - Violino
15. click	id=duration	
16. type	id=duration	11:11
17. click	id=addLesson	
18. assert text	id=messErr	Titolo già grave esistente

Command

Target

Value

Description

Log Reference

16. type on id=duration with value 11:11 OK 15:58:28

17. click on id=addLesson OK 15:58:28

18. assertText on id=messErr with value Titolo già grave esistente OK 15:58:28

TCS43 completed successfully 15:58:28

Nome test case:	TCS45
Data e ora:	16/01/20 12:21
Tester:	Claudia Buono
Output atteso:	L'inserimento va a buon fine poiché tutti i campi sono stati inseriti correttamente.
Output reale:	L'inserimento va a buon fine poiché tutti i campi sono stati inseriti correttamente.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Esistenza, Formato, Lunghezza

Selenium IDE - StudyMeSelenium*


Project: StudyMeSelenium*



Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
8	click	css=fa-plus-square	
9	click	id=url	
10	type	id=url	https://www.youtube.com/embed/2OIAvMeNP6E
11	click	id=title	
12	type	id=title	prima elementare
13	click	id=duration	
14	type	id=duration	11:11
15	click	id=addLesson	
16	assert text	id=TitoloLezione	prima elementare

Command # 

Target  

Value

Description

Log Reference

14. type on id=duration with value 11:11 OK 12:21:57

15. click on id=addLesson OK 12:21:57

16. assertText on id=TitoloLezione with value prima elementare OK 12:21:58

***TCS45* completed successfully** 12:21:58

9.2 Modifica Lezione

Nome test case:	TCS46
Data e ora:	15/01/20 16:36
Tester:	Claudia Buono
Output atteso:	La modifica non va a buon fine poiché il campo non rispetta la lunghezza del nome.
Output reale:	La modifica non va a buon fine poiché il campo non rispetta la lunghezza del nome.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Lunghezza

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
6 mouse up at	css=container	1136,267
7 click	css=pacchetto:nth-child(4) a	
8 mouse over	css=lezione:nth-child(1) .modificaLezione	
9 click	css=lezione:nth-child(1) .modificaLezione	
10 mouse out	css=lezione:nth-child(1) .modificaLezione	
11 click	id=nomeModificaLezione	
12 type	id=nomeModificaLezione	a
13 click	id=updateButton	
14 assert text	id=insuccess	Inserire un titolo compreso tra i 5 e 35 caratteri

Command

Target

Value

Description

Log Reference

12. type on id=nomeModificaLezione with value a OK 16:36:02

13. click on id=updateButton OK 16:36:02

14. assertText on id=insuccess with value Inserire un titolo compreso tra i 5 e 35 caratteri OK 16:36:02

TCS46* completed successfully 16:36:02

Nome test case:	TCS47
Data e ora:	15/01/20 12:27
Tester:	Claudia Buono
Output atteso:	La modifica va a buon fine poiché tutte le condizioni sono verificate.
Output reale:	La modifica va a buon fine poiché tutte le condizioni sono verificate.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Esistenza, Formato, Lunghezza

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests... Q

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
TCS39	7	click	css=pageNewDomElement
TCS40	8	run script	window.scrollTo(0,212)
TCS41	9	click	css=lezione:nth-child(1).modificaLezione
TCS42	10	click	id=nomeModificaLezione
TCS43	11	type	id=nomeModificaLezione corso matematica bimbi
TCS45	12	click	id=updateButton
TCS46	13	mouse over	css=lezione:nth-child(1).modificaLezione
TCS47*	14	assert text	id=TitoloLezione corso matematica bimbi

Command

Target

Value

Description

Log Reference

12. click on id=updateButton OK 12:27:42

13. mouseOver on css=lezione:nth-child(1).modificaLezione OK 12:27:42

14. assertText on id=TitoloLezione with value corso matematica bimbi OK 12:27:43

TCS47 completed successfully 12:27:43

Nome test case:	TCS49
Data e ora:	15/01/20 16:45
Tester:	Claudia Buono
Output atteso:	La modifica non va a buon fine poiché il campo non rispetta le condizioni dell'URL.
Output reale:	La modifica non va a buon fine poiché il campo non rispetta le condizioni dell'URL.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
5 mouse move at	css=.container	1143,328
6 mouse up at	css=.container	1143,328
7 click	css=.pacchetto:nth-child(4) a	
8 click	css=.lezione:nth-child(1) .modificaLezione	
9 click	id=urlModificaLezione	
10 type	id=urlModificaLezione	i
11 click	css=#updateUrl > #updateButton	
12 assert text	id=insuccess	Url non valido!

Command

Target

Value

Description

Log Reference

10. type on id=urlModificaLezione with value i OK 16:45:12

11. click on css=#updateUrl > #updateButton OK 16:45:12

12. assertText on id=insuccess with value Url non valido! OK 16:45:12

***TCS49* completed successfully** 16:45:12

Nome test case:	TCS50
Data e ora:	15/01/20 16:45
Tester:	Claudia Buono
Output atteso:	La modifica non va a buon fine poiché l'URL è già presente nel database.
Output reale:	La modifica non va a buon fine poiché l'URL è già presente nel database.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Esistenza

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

	Command	Target	Value
TCS39	4 mouse down at	css=.container	1140,242
TCS40	5 mouse move at	css=.container	1140,242
TCS41	6 mouse up at	css=.container	1140,242
TCS42	7 click	css=.pacchetto:nth-child(4) a	
TCS43	8 click	css=.lezione:nth-child(1) .modificaLezione	
TCS46	9 click	id=urlModificaLezione	
TCS49	10 type	id=urlModificaLezione	https://www.youtube.com/embed/KYjwcntuEKk
TCS50*	11 click	css=#updateUri > #updateButton	
	12 assert text	id=insuccess	Uri già esistente

Untitled

Command

Target

Value

Description

Log Reference

10. type on id=urlModificaLezione with value https://www.youtube.com/embed/KYjwcntuEKk OK 16:59:48

11. click on css=#updateUri > #updateButton OK 16:59:48

12. assertText on id=insuccess with value Uri già esistente OK 16:59:48

TCS50* completed successfully 16:59:48

9.3 Inserimento recensione

Nome test case:	TCS53
Data e ora:	15/01/20 17:13
Tester:	Claudia Buono
Output atteso:	La creazione della recensione non va a buon fine poiché il testo del titolo non spetta i criteri di lunghezza.
Output reale:	La creazione della recensione non va a buon fine poiché il testo del titolo non spetta i criteri di lunghezza.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Lunghezza

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
4 mouse down at	css=.container	1128,256
5 mouse move at	css=.container	1128,256
6 mouse up at	css=.container	1128,256
7 click	css=sottocatnth-child(8).catalogo	
8 click	id=recensione	
9 click	id=titoloR	
10 type	id=titoloR	o
11 click	css=bottoneDefaultnth-child(3)	
12 assert text	id=insuccess	Il titolo deve essere tra i 5 e i 30 caratteri

TCS39

TCS40

TCS41

TCS42

TCS43

TCS46

TCS49

TCS50

TCS53*

Untitled

Command

Target

Value

Description

Log Reference

10. type on id=titoloR with value o OK 17:13:10

11. click on css=bottoneDefaultnth-child(3) OK 17:13:10

12. assertText on id=insuccess with value Il titolo deve essere tra i 5 e i 30 caratteri OK 17:13:10

TCS53 completed successfully 17:13:11

Nome test case:	TCS54
Data e ora:	15/01/20 17:13
Tester:	Claudia Buono
Output atteso:	La creazione della recensione non va a buon fine poiché il campo del commento non rispetta i criteri di lunghezza.
Output reale:	La creazione della recensione non va a buon fine poiché il campo del commento non rispetta i criteri di lunghezza.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Formato

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/SourceHomePage.jsp

Command	Target	Value
6 mouse up at	css=.container	1135,249
7 click	css=sottocat:nth-child(8).catalogo	
8 click	id=recensione	
9 click	id=titoloR	
10 type	id=titoloR	bellissima
11 click	id=btRecensione	
12 type	id=btRecensione	si
13 click	css=bottoneDefault:nth-child(3)	
14 assert text	id=insuccess	Il testo deve essere tra i 10 e i 30 caratteri

Command

Target

Value

Description

Log Reference

12. type on id=btRecensione with value si OK 17:21:56

13. click on css=bottoneDefault:nth-child(3) OK 17:21:56

14. assertText on id=insuccess with value Il testo deve essere tra i 10 e i 30 caratteri OK 17:21:56

TCS54* completed successfully 17:21:56

Nome test case:	TCS55
Data e ora:	15/01/20 17:13
Tester:	Claudia Buono
Output atteso:	La creazione della recensione non va a buon fine poiché il campo del titolo e del commento sono vuoti.
Output reale:	La creazione della recensione non va a buon fine poiché il campo del titolo e del commento sono vuoti.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Esistenza

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/Source/HomePage.jsp

Command	Target	Value
2 set window size	1165x728	
3 click	css=a:nth-child(6).sfondo	
4 mouse down at	css=container	1141,217
5 mouse move at	css=container	1141,217
6 mouse up at	css=container	1141,217
7 click	css=sottocat:nth-child(8).catalogo	
8 click	id=recensione	
9 click	css=bottoneDefault:nth-child(3)	
10 assert text	id=insuccess	Tutti i parametri devono essere passati

Command

Target

Value

Description

Log Reference

8. click on id=recensione OK 17:33:29

9. click on css=bottoneDefault:nth-child(3) OK 17:33:29

10. assertText on id=insuccess with value Tutti i parametri devono essere passati OK 17:33:30

TCS55* completed successfully 17:33:30

Nome test case:	TCS56
Data e ora:	15/01/20 17:13
Tester:	Claudia Buono
Output atteso:	La creazione della recensione viene effettuata con successo.
Output reale:	La creazione della recensione viene effettuata con successo.
Esito del test:	Corretto
Numero prove:	1
Categoria:	Esistenza, Formato, Lunghezza

Selenium IDE - StudyMeSelenium*

Project: StudyMeSelenium*

Tests +

Search tests...

http://localhost:8080/SourceHomePage.jsp

	Command	Target	Value
TCS39	6 mouse up at	css=.container	1365,160
TCS40	7 click	css=pacchetto:nth-child(4).catalogo	
TCS41	8 click	id=recensione	
TCS42	9 click	id=titoloR	
TCS43	10 type	id=titoloR	recensione
TCS45	11 click	id=bdRecensione	
TCS46	12 type	id=bdRecensione	bella ed emozionante
TCS47	13 click	css=bottoneDefault:nth-child(3)	
TCS47	14 assert text	id=recensore	Adele

Command

Target

Value

Description

TCS56*

Log Reference

12. type on id=bdRecensione with value bella ed emozionante OK 15:25:08

13. click on css=bottoneDefault:nth-child(3) OK 15:25:08

14. assertText on id=recensore with value Adele OK 15:25:08

TCS56 completed successfully 15:25:09